

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior



SISTEMA AUTOMÁTICO DE DETECCIÓN Y ETIQUETADO DE CARAS EN IMÁGENES

Proyecto Fin de Carrera
Ingeniería de Telecomunicación
Especialidad Sonido e Imagen

Autor: Tamara Herrero Vez
Tutor: Julio Villena Román

Madrid, Marzo 2010

Título: Sistema automático de detección y etiquetado de caras en imágenes.

Autor: Tamara Herrero Vez

Tutor: Julio Villena Román

EL TRIBUNAL

Presidente:

M. Carmen Fernández Panadero

Secretario:

Raquel M. Crespo

Vocal:

Jaime de Uriarte

Realizado el acto de defensa del Proyecto Fin de Carrera el día 23 de Marzo de 2010 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo. Vocal

Fdo: Presidente

Fdo: Secretario

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

AGRADECIMIENTOS

Me gustaria dedicar este proyecto a todas las personas que han estado de una manera o de otra en mi vida.

A mis padres, por exigirme lo que sabian que podia dar y por animarme a seguir adelante cada dia.

En especial a mi novio, Sergio, porque esta carrera nos unió... y porque él es el responsable de mi felicidad. Gracias por aguantarme en estos meses de verdadero infierno, sin tu ayuda, sabes que no habria podido terminar este proyecto.

A Maria que ya desde el colegio me ha acompañado... Espero que dentro de muy poco seas tu quien escriba algo bonito sobre mi en tu memoria, porque yo se, más que nadie, que tu puedes...

A Lorena, porque su persistencia y su capacidad para continuar son tan sorprendentes, que sé que al final, nos graduaremos juntas;;

A Dani, porque aunque "le conocí ayer", podria decirse que la mitad de mi carrera es suya. Tetilla ☺

A Marian, la niña, la delega, porque veras que tus esfuerzos dan su fruto muy pronto...y este verano a celebrarlo ;

A Patri, porque espero que esta dedicatoria la haga un poquito más feliz, porque como puedes ver, al final todo va saliendo bien poco a poco... Sólo hay que tener paciencia.

A todas las personas que conocí dia tras dia en la universidad y de las cuales me llevo un gran recuerdo: Irene, Jc, Diego, Kike, Lola, Ana, los insociables.... Debo ser la única persona del mundo a la que se le acaba la hoja de dedicatorias, ojala os pudiera decir a todos algo más.

Gracias tambien a todos mis profesores y en especial a mi tutor, sin el cual este proyecto nunca podria haber sido realizado.

Como no podia ser de otra manera, le dedico este proyecto a la persona más importante de mi vida, **mi hermana**. Porque ella me ha enseñado lo que es luchar por lo que se quiere y porque de alguna manera le tengo que compensar todo lo que hace cada dia. Gracias, porque para mi no hay mejor regalo, que qué estes hoy aquí conmigo.

Y como mencion especial a mi tio, porque ojala pudiera haberme visto llegar hasta aquí.

A todos, MUCHAS GRACIAS;;

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

RESUMEN

En este proyecto se presenta un sistema automático de detección y etiquetado de imágenes estáticas.

Detectar rostros en fotografías es un problema en pleno auge en el campo de la visión por ordenador y son muchas las aplicaciones prácticas que se tienen en la actualidad, videoconferencia, vigilancia, control de acceso, etc. Además el etiquetado de fotos es un recurso muy necesitado en Internet, donde la cantidad de imágenes crece exponencialmente y es una necesidad cada vez mayor el poder clasificarlas de alguna manera.

El objetivo de este sistema es devolver la imagen con todas las caras detectadas y descritas por una serie de etiquetas. El enfoque adoptado en este proyecto se puede dividir en dos fases, la de detección de la cara en la imagen y la de extracción de características para su etiquetado.

La fase de detección, a su vez, se ha realizado evaluando dos algoritmos basados en diferentes técnicas para su comparación. Por un lado se tiene una detección de bajo nivel, donde segmentamos la imagen en función de los píxeles de piel. Posteriormente se realiza una selección de regiones de piel candidatas a ser caras, que si cumplen unas determinadas condiciones, pasarán a ser la cara detectada. Por el otro lado, se tiene otra técnica basada en plantillas, donde se crea una función de probabilidad de piel en función a una base de datos de pieles. Tras esto, se emplea una plantilla de una cara robotizada, la cual se correlaciona con las regiones de piel, para obtener la cara detectada.

En la fase de etiquetado de la imagen, se tiene ya cada cara seleccionada y se empieza a trabajar a partir de ella. Empleando funciones básicas de MATLAB, se van obteniendo características que ayudarán a etiquetar las caras de dichas imágenes, según su raza, su color de ojos, sexo, etc.

En la evaluación del sistema implementado se ve la gran influencia del tipo de imágenes utilizadas, siendo mejores los resultados cuando las imágenes cumplen unas ciertas condiciones de tamaño, de luminosidad, de orientación, etc. Además es muy importante la precisión del detector de caras para un correcto etiquetado.

ÍNDICE

1. INTRODUCCIÓN.....	1
1.1 MOTIVACIÓN.....	1
1.2 OBJETIVOS.....	2
1.3 ESTRUCTURA DE LA MEMORIA.....	3
2. ESTADO DEL ARTE.....	5
2.1 CONCEPTOS GENERALES.....	5
2.1.1 LA IMAGEN DIGITAL.....	6
2.1.2 ESPACIOS DE COLOR.....	7
2.1.2.1 Espacio de color RGB.....	8
2.1.2.2 Espacio de color YCbCr.....	9
2.1.2.3 Espacio de color HSV.....	9
2.1.3 PROCESAMIENTO DE LA IMAGEN.....	10
2.1.3.1 Operaciones puntuales.....	10
2.1.3.1.1 Modificación del contraste.....	11
2.1.3.1.2 Modelado del histograma.....	12
2.1.3.1.3 Operaciones entre imágenes.....	13
2.1.3.2 Operaciones espaciales.....	13
2.1.3.2.1 Filtro paso bajo.....	14
2.1.3.2.2 Filtro gaussiano.....	14
2.1.3.2.3 Filtro paso alto.....	15
2.1.3.2.4 Filtros detectores de bordes.....	15
2.1.3.3 Filtrado en frecuencia.....	17
2.1.3.3.1 Filtros morfológicos.....	17
2.1.3.3.2 Filtros de textura.....	19
2.1.4 SEGMENTACIÓN.....	20
2.1.4.1 Técnicas basadas en los bordes.....	20
2.1.4.2 Umbralización.....	21
2.1.4.3 Técnicas de agrupamiento o clustering.....	22
2.1.5 REPRESENTACIÓN Y DESCRIPCIÓN.....	23
2.1.5.1 Representación.....	23
2.1.5.2 Descripción.....	25
2.1.5.2.1 Descriptores de forma.....	25
2.1.5.2.2 Descriptores de región.....	25
2.1.5.2.3 Componentes principales, PCA.....	25
2.1.6 RECONOCIMIENTO.....	26
2.1.6.1 Métodos de decisión teórica.....	26
2.1.6.1.1 Adaptación.....	26
2.1.6.1.2 Redes Neuronales.....	27
2.2 DETECCIÓN DE CARAS.....	28
2.2.1. TÉCNICAS BASADAS EN RASGOS FACIALES.....	30
2.2.1.1. Análisis de bajo nivel.....	30
2.2.1.1.1. Detección de bordes.....	30
2.2.1.1.2. Información de grises.....	31
2.2.1.2. Análisis de rasgos.....	31

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

2.2.1.2.1. Búsqueda de rasgos.....	31
2.2.1.2.2. Análisis de constelaciones	32
2.2.1.3. Análisis de formas activas.....	32
2.2.2. TÉCNICAS BASADAS EN LA IMAGEN.....	33
2.2.2.1. Métodos basados en subespacios.....	33
2.2.2.2. Redes neuronales.....	33
2.2.2.3. Métodos estadísticos.....	34
2.3 ETIQUETADO DE IMÁGENES.....	34
2.3.1 CLEF.....	35
2.3.2 PASCAL VOC.....	39
3. DISEÑO E IMPLEMENTACION.....	41
3.1 ARQUITECTURA DEL SISTEMA.....	41
3.1.1. HERRAMIENTAS UTILIZADAS	42
3.2. BASES DE DATOS.....	43
3.2.1. BASE DE DATOS A: FOTOGRAFIAS	43
3.2.2. BASE DE DATOS B: FOTOGRAFIAS DE CARAS.....	45
3.2.3. BASE DE DATOS C: TIPOS DE PIEL	46
3.2.3. BASE DE DATOS E: PLANTILLAS DE OJOS	47
3.2.4. BASE DE DATOS D: TIPOS DE PELO	47
3.3. FASES DEL PROYECTO.....	48
3.4. PREPROCESADO DE LAS IMÁGENES.....	48
3.4.1. ELIMINACION DEL RUIDO.....	49
3.4.1.1. Filtros de paso bajo.....	49
3.4.1.2. Filtro de medianas	50
3.5. DETECCIÓN DE CARAS.....	52
3.5.1. DETECCIÓN DE CARA DE BAJO NIVEL.....	52
3.5.1.1 Detección de píxeles de piel.....	52
3.5.1.2. Filtrado y agrupamiento	55
3.5.1.3. Validación de caras.....	57
3.5.2. DETECCIÓN DE CARA CON PLANTILLAS.....	59
3.5.2.1 Detección de píxeles de piel.....	59
3.5.2.2. Filtrado y agrupamiento	61
3.5.2.3. Validación de caras.....	62
3.6. ETIQUETADO DE CARAS.....	64
3.6.1. RAZA.....	66
3.6.2. COLOR DE OJOS	69
3.6.3. COLOR DE PELO.....	72
3.6.4. BARBA	74
3.6.5. LONGITUD DEL PELO.....	75
3.6.6. SEXO.....	76
4. EVALUACIÓN DEL SISTEMA.....	81
4.1 FASE DE DETECCIÓN.....	82
4.1.1 SELECCIÓN DE LOS OPERADORES MORFOLÓGICOS.....	83
4.1.2 CAMBIO DE TAMAÑO DE IMAGEN DE ENTRADA.....	85
4.1.3 CAMBIO DE UMBRALES DE DETECCIÓN	87
4.1.4 RESUMEN DE LOS ALGORITMOS.....	90
4.1.4.1 Evaluación en función del n° de personas.....	91

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

4.1.4.2 Evaluación en función de la raza.....	93
4.1.4.3 Evaluación en función del sexo.....	94
4.2 FASE DE ETIQUETADO.....	95
4.2.1 CAMBIO DE UMBRALES.....	96
4.2.1.1 Factor de forma en etiquetado de ojos.....	96
4.2.1.2 Tamaño del área en etiquetado de ojos.....	97
4.2.1.3 Factor de forma en etiquetado de sexo.....	97
4.2.1.4 Relación de aspecto en etiquetado de sexo.....	98
4.2.2 RESUMEN DEL ALGORITMO DE ETIQUETADO.....	98
4.2.2.1 Raza.....	99
4.2.2.2 Color de ojos.....	100
4.2.2.3 Color de pelo.....	101
4.2.2.4 Barba.....	102
4.2.2.5 Longitud del pelo.....	103
4.2.2.6 Sexo.....	104
4.2.2.7 Resumen del conjunto de etiquetado.....	105
4.3 FASE FINAL:EVALUACIÓN DEL SISTEMA GLOBAL.....	107
4.3.1 DETECCIÓN DE BAJO NIVEL Y ETIQUETADO.....	107
4.3.1.1 Raza.....	108
4.3.1.2 Color de ojos.....	109
4.3.1.3 Color de pelo.....	110
4.3.1.4 Barba.....	112
4.3.1.5 Longitud de pelo.....	113
4.3.1.6 Sexo.....	113
4.3.2 DETECCIÓN BASADA EN PLANTILLAS Y ETIQUETADO.....	114
4.3.2.1 Raza.....	115
4.3.2.2 Color de ojos.....	116
4.3.2.3 Color de pelo.....	117
4.3.2.4 Barba.....	118
4.3.2.5 Longitud del pelo.....	119
4.3.2.6 Sexo.....	120
4.3.3 COMPARACIÓN ENTRE MÉTODOS CON ETIQUETADO.....	121
4.4 ANALISIS DEL ERROR.....	126
4.4.1 FASE DE DETECCIÓN.....	126
4.4.1 FASE DE ETIQUETADO.....	130
5. CONCLUSIONES Y TRABAJOS FUTUROS.....	135
5.1 CONCLUSIONES.....	135
5.2 LINEAS FUTURAS DE TRABAJO.....	137
6. ANEXOS.....	141
6.1. DETECCIÓN DE CARAS DE BAJO NIVEL.....	141
6.2 DETECCIÓN DE CARAS BASADA EN PLANTILLAS.....	142
6.3 FASE DE ENTRENAMIENTO DEL ETIQUETADO.....	143
6.4 DETECCIÓN DE BAJO NIVEL CON ETIQUETADO.....	145
6.5 DETECCIÓN BASADA EN PLANTILLAS CON ETIQUETADO.....	148
6.5 MANUAL DEL USUARIO.....	150
6.6 CÓDIGO MATLAB.....	151
7. BIBLIOGRAFIA.....	175

ÍNDICE DE FIGURAS

Figura 2.1:	Diagrama de bloques del análisis de imágenes.....	5
Figura 2.2:	a) Imagen original.....	6
	b) Estructura matricial de la imagen.....	6
Figura 2.3:	a) Imagen en color.....	7
	b) Imagen en escala de gris.....	7
	c) Imagen binaria.....	7
Figura 2.4:	Cubo de coordenadas RGB.....	8
Figura 2.5:	Cilindro de coordenadas de HSV.....	9
Figura 2.6:	Operación puntual de un pixel.....	10
Figura 2.7:	Funciones para la modificación del contraste.....	11
Figura 2.8:	a) Imagen original.....	11
	b) Umbralización siendo $a=b=100$	11
Figura 2.9:	a) Imagen original.....	12
	b) Histograma de la imagen.....	12
	c) Histograma ecualizado de la imagen.....	12
Figura 2.10:	Operación espacial de un pixel.....	13
Figura 2.11:	Máscara de convolución de 3×3	14
Figura 2.12:	a) Imagen con ruido $N(0, 0.03)$	15
	b) Imagen filtrada con un filtro gaussiano 3×3	15
Figura 2.13:	Máscara de convolución de 3×3 Kernel.....	15
Figura 2.14:	a) Sobel.....	16
	b) Laplaciano.....	16
	c) Canny.....	16
Figura 2.15:	a) Imagen binaria.....	17
	b) Imagen dilatada.....	17
Figura 2.16:	a) Imagen binaria.....	18
	b) Imagen erosionada.....	18
Figura 2.17:	a) Imagen binaria.....	19
	b) Imagen tras la apertura.....	19
	c) Imagen tras el cierre.....	19
Figura 2.17:	Ejemplo de Transformada de Hough.....	21
Figura 2.18:	a) Imagen en escala de gris.....	22
	b) Histograma.....	22
	c) Imagen binaria resultante con umbral=120.....	22
	d) Imagen binaria resultante con umbral=180.....	22
Figura 2.19:	Ejemplo de clustering.....	22
Figura 2.20:	a) Contorno original.....	24
	b) Remuestreo.....	24
	c) Código de cadena con conectividad a 4.....	24
	d) Código de cadena con conectividad a 8.....	24
Figura 2.21:	Ejemplo de técnica de fusión.....	24
Figura 2.22:	Ejemplos de firmas distancia-ángulo.....	24
Figura 2.23:	Perceptrón para dos clases de patrones.....	27
Figura 2.24:	Perceptrón multicapa progresivo.....	27
Figura 2.25:	Jerarquía de clases de la detección visual.....	39
Figura 3.1:	Diagrama de bloques del sistema.....	41
Figura 3.2:	Ejemplo de imágenes de la base de datos A.....	44
Figura 3.3:	Ejemplo de imágenes de la base de datos B.....	46
Figura 3.4:	Ejemplos de imágenes de la base de datos C.....	46
Figura 3.5:	Ejemplo de imágenes de la base de datos D.....	47
Figura 3.6:	Imágenes de la base de datos E.....	47

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Figura 3.7:	Máscara de filtrado de 3x3.....	49
Figura 3.8:	a) Imagen de la base de datos A.....	50
	b) Imagen segmentada con detector de plantilla...	50
	c) Imagen filtrada con el filtro paso bajo.....	50
Figura 3.9:	Funcionamiento del filtro de medianas.....	50
Figura 3.10:	a) Imagen de la base de datos A.....	51
	b) Imagen segmentada con detector de bajo nivel..	51
	c) Imagen filtrada con el filtro de mediana.....	51
Figura 3.11:	a) c) Imágenes de la base de datos A.....	54
	b) d) Imagen binaria detector de bajo nivel.....	54
Figura 3.12:	a) d) Imágenes segmentadas.....	56
	b) e) Imágenes tras la erosión de disco 3.....	56
	c) f) Imágenes tras la dilatación de disco 3.....	56
Figura 3.13:	a) d) Imágenes de la base de datos A.....	60
	b) e) Imagen de probabilidad de piel.....	60
	c) f) Imagen binaria detector de plantilla.....	60
Figura 3.14:	a) d) Imágenes segmentadas.....	62
	b) e) Imágenes tras la erosión de disco 10.....	62
	c) f) Imágenes tras la dilatación de disco 8.....	62
Figura 3.15:	a) Región que ha pasado los test.....	63
	b) Plantilla modificada en función de la región..	63
	c) Cara detectada.....	63
Figura 3.16:	a) c) Recortes ampliados de piel.....	67
	b) d) Histograma de Y.....	67
Figura 3.17:	a) c) Cara con recorte de piel.....	68
	b) d) Histograma de Y.....	68
Figura 3.18:	a) Imagen con cara recuadrada a mano.....	70
	b) Segmentación en función del espacio RGB.....	70
	c) Imagen inversa.....	70
Figura 3.19:	Candidatos a ojo de la figura 3.17.....	70
Figura 3.20:	a) Plantillas de ojo derecho azul.....	71
	b) Plantilla ojo izquierdo negro.....	71
	c) Plantilla ojo derecho marrón.....	71
	c) Plantilla ojo derecho marrón.....	71
Figura 3.21:	a) Imagen con cara detectada.....	72
	b) Zona de pelo recortado.....	72
Figura 3.22:	a) Imagen con la cara detectada a mano.....	74
	b) Zona de la barbilla recortada.....	74
Figura 3.23:	a) c) Imagen con la región de pelo recuadrada....	75
	b) d) Imagen recortada de la zona de pelo.....	75
Figura 3.24:	a) Imagen con la cara recuadrada.....	79
	b) Pelo recortado.....	79
	c) Zona de la barbilla recortada.....	79
	d) e) Zonas de pelo largo recortadas.....	79
	f) Resultados del etiquetado de sexo.....	79
Figura 4.1:	a) Imagen con tamaño (351x172).....	85
	b) Imagen con tamaño (631x313).....	85
	c) Caras detectadas de la imagen a).....	85
	d) Caras detectadas de la imagen b).....	85
Figura 4.2:	a) Imagen con tamaño (351x172).....	86
	b) Imagen con tamaño (631x313).....	86
	c) Caras detectadas de la imagen a).....	86
	d) Caras detectadas de la imagen b).....	86
Figura 4.3:	a) Imagen "cara39.jpg" base datos B.....	100
	b) Imagen "cara40.jpg" base de datos C.....	100

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Figura 4.4:	a) Imagen "32.jpg" de la base de datos A.....	110
	b) Resultado del etiquetado de color de pelo....	110
Figura 4.5:	Diagrama de barras de las etiquetas.....	123
Figura 4.6:	a) Imagen original.....	126
	b) Imagen tras el filtrado.....	126
	c) Imagen con la cara detectada.....	126
Figura 4.7:	a) Imagen original.....	127
	b) Imagen tras el filtrado.....	127
	c) Imagen con la cara detectada.....	127
Figura 4.8:	a) Imagen original.....	127
	b) Imagen tras el filtrado.....	127
	c) Imagen con la cara detectada.....	127
Figura 4.9:	a) Imagen original.....	128
	b) Imagen tras el filtrado.....	128
	c) Imagen con la cara detectada.....	128
Figura 4.10:	a) Imagen original.....	128
	b) Imagen tras el filtrado.....	128
	c) Imagen con la cara detectada.....	128
Figura 4.11:	a) Imagen original.....	129
	b) Imagen tras el filtrado.....	129
	c) Imagen con la cara detectada.....	129
Figura 4.12:	a) Imagen original.....	129
	b) Imagen tras el filtrado.....	129
	c) Imagen con la cara detectada.....	129
Figura 4.13:	a) Imagen original.....	130
	b) Imagen tras el filtrado.....	130
	c) Imagen con la cara detectada.....	130
Figura 4.14:	a) Imagen con la cara detectada.....	131
	b) Cara ampliada de la zona de la frente.....	131
Figura 4.15:	a) Imagen con la cara detectada.....	131
	b) Cara ampliada de la zona de la frente.....	131
Figura 4.16:	a) Imagen con la cara detectada.....	132
	b) Zona del pelo ampliada.....	132
Figura 4.17:	a) Imagen con la cara detectada.....	132
	b) Zona de pelo ampliada.....	132
Figura 4.18:	a) Imagen con la cara detectada.....	133
	b) Cara detectada.....	133
	c) Zona de la barbilla ampliada.....	133
Figura 4.19:	a) Imagen con la cara detectada.....	133
	b) Recorte ampliado para la longitud de pelo....	133
	c) Recorte ampliado para la longitud de pelo....	133

ÍNDICE DE TABLAS

Tabla 2.1:	Taxonomía de detección de caras	29
Tabla 2.2:	Evolución CLEF	36
Tabla 3.1:	Funciones de Matlab para el filtrado	51
Tabla 3.2:	Rendimiento de diferentes detector de piel	53
Tabla 3.3:	Funciones de MATLAB para el filtrado	57
Tabla 3.4:	Funciones de MATLAB para validación de caras	58
Tabla 3.5:	Funciones de MATLAB para la segmentación	61
Tabla 3.6:	Funciones de MATLAB para validación de caras	64
Tabla 3.7:	Posibles valores del etiquetado de raza	68
Tabla 3.8:	Funciones de MATLAB para etiquetado de raza	69
Tabla 3.9:	Posibles valores del etiquetado de color de ojos ..	71
Tabla 3.10:	Posibles valores para el color de pelo	73
Tabla 3.11:	Funciones de MATLAB para el color de pelo	73
Tabla 3.12:	Posibles valores del etiquetado de barba	74
Tabla 3.13:	Posibles valores para la longitud de pelo	75
Tabla 3.14:	Posibles valores del etiquetado de sexo	77
Tabla 3.15:	Tabla de la verdad para el etiquetado de sexo	79
Tabla 4.1:	Matriz de confusión	81
Tabla 4.2:	Operadores morfológicos detección de bajo nivel ...	83
Tabla 4.3:	Operadores morfológicos detección con plantillas ..	84
Tabla 4.4:	Cambio de tamaño detección de bajo nivel	86
Tabla 4.5:	Cambio de tamaño detección basado en plantillas ...	87
Tabla 4.6:	Factor de forma detección de bajo nivel	87
Tabla 4.7:	Relación de aspecto en la detección de bajo nivel ..	88
Tabla 4.8:	Solidez detección de bajo nivel	88
Tabla 4.9:	Relación de aspecto para detección plantillas	89
Tabla 4.10:	Correlación detección basada en plantillas	89
Tabla 4.11:	Comparación entre los algoritmos	90
Tabla 4.12:	Evaluación del nº de personas bajo nivel	92
Tabla 4.13:	Evaluación del nº de personas plantillas	92
Tabla 4.14:	Evaluación en función de la raza bajo nivel	93
Tabla 4.15:	Evaluación en función de la raza plantillas	94
Tabla 4.16:	Evaluación en función del sexo bajo nivel	94
Tabla 4.17:	Evaluación en función del sexo plantillas	95
Tabla 4.18:	Pruebas en el factor de forma	96
Tabla 4.19:	Prueba de tamaño del área	97
Tabla 4.20:	Prueba del factor de forma en etiquetado de sexo ..	98
Tabla 4.21:	Prueba de la relación de aspecto para el sexo	98
Tabla 4.22:	Matriz de confusión del etiquetado de Raza	99
Tabla 4.23:	Evaluación para el etiquetado de raza	100
Tabla 4.24:	Matriz de confusión del del color de ojos	100
Tabla 4.25:	Evaluación para el etiquetado de color de ojos ...	101
Tabla 4.26:	Matriz de confusión del color de pelo	102
Tabla 4.27:	Evaluación para el etiquetado del color de pelo ..	102
Tabla 4.28:	Matriz de confusión para el etiquetado de barba ..	103
Tabla 4.29:	Evaluación para el etiquetado de barba	103
Tabla 4.30:	Matriz de confusión de la longitud del pelo	103
Tabla 4.31:	Evaluación del etiquetado de la longitud del pelo	104
Tabla 4.32:	Matriz de confusión para el etiquetado se sexo ...	104
Tabla 4.33:	Evaluación para el etiquetado de sexo	104

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Tabla 4.34: Evaluación para el conjunto de etiquetado.....	105
Tabla 4.35: Evaluación de etiquetado.....	106
Tabla 4.36: Matriz de confusión para el etiquetado de raza...	108
Tabla 4.37: Evaluación del etiquetado de raza.....	108
Tabla 4.38: Matriz de confusión de color de ojos.....	109
Tabla 4.39: Evaluación del etiquetado de color de ojos.....	109
Tabla 4.40: Matriz de confusión de color de pelo.....	111
Tabla 4.41: Evaluación del etiquetado del color de pelo.....	111
Tabla 4.42: Matriz de confusión del etiquetado de barba.....	112
Tabla 4.43: Evaluación del etiquetado de barba.....	112
Tabla 4.44: Matriz de confusión de longitud de pelo.....	113
Tabla 4.45: Evaluación del etiquetado de longitud de pelo....	113
Tabla 4.46: Matriz de confusión del etiquetado de sexo.....	114
Tabla 4.47: Evaluación del etiquetado de sexo.....	114
Tabla 4.48: Matriz de confusión del etiquetado de raza.....	115
Tabla 4.49: Evaluación del etiquetado de raza.....	115
Tabla 4.50: Matriz de confusión de color de ojos.....	116
Tabla 4.51: Evaluación del etiquetado de color de ojos.....	117
Tabla 4.52: Matriz de confusión del color de pelo.....	117
Tabla 4.53: Evaluación del etiquetado de color de pelo.....	118
Tabla 4.54: Matriz de confusión del etiquetado de barba.....	118
Tabla 4.55: Evaluación del etiquetado de barba.....	119
Tabla 4.56: Matriz de confusión de longitud de pelo.....	120
Tabla 4.58: Matriz de confusión del etiquetado de sexo.....	120
Tabla 4.59: Evaluación del etiquetado de sexo.....	121
Tabla 4.60: Evaluación etiquetas con detección de bajo nivel.	122
Tabla 4.61: Evaluación etiquetas detección plantillas.....	122
Tabla 4.62: Precisión media de todas las etiquetas.....	123
Tabla 4.63: N° imágenes etiquetas correctas de bajo nivel....	124
Tabla 4.64: N° imágenes etiquetas correctas plantillas.....	125
Tabla 4.65: Evaluación del n° de etiquetas correctas.....	125

1. INTRODUCCIÓN

1.1 MOTIVACIÓN

Desde las últimas décadas, el proceso tecnológico ha avanzado de tal manera, que tareas que eran realizadas tradicionalmente por seres humanos, son ahora realizadas por sistemas automatizados que optimizan el tiempo de ejecución. Dentro de la amplia gama de posibles actividades que pueden automatizarse, aquella relacionada con la capacidad para establecer la identidad de los individuos ha cobrado una gran importancia y como consecuencia directa, la Biometría Informática se ha transformado en un área emergente.

La Biometría Informática es la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo para su clasificación. Una de las metas más importantes que persiguen los investigadores es comprender el complejo sistema visual humano, que logra almacenar la información de rostros y utilizarla para discriminar identidades aun con condiciones adversas.

El reconocimiento de caras es un sistema biométrico que tiene importantes ventajas sobre otros sistemas, al ser un método no intrusivo (la persona no siente invadida su intimidad), no se necesitan acciones de identificación ni someterse a ningún tipo de análisis y además no se utilizan códigos que se tengan que memorizar.

También existen inconvenientes, ya que la detección automática de caras solo funciona cuando los sujetos se muestran colaborativos, sin enmascarar su cara y en condiciones favorables de luminosidad, posición de la persona, etc.

Un campo importante en el desarrollo de los sistemas biométricos es el de la recuperación de información, cuyo área más activa desde los años 70, ha sido concretamente la recuperación de imágenes.

En general, la recuperación de imágenes puede realizarse tanto basada en texto, que requiere una descripción manual de la imagen, como basada en contenido, que utiliza técnicas automáticas.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

La recuperación de imágenes basada en texto, suele ser ambigua, incompleta y consume mucho tiempo, mientras que la recuperación basada en contenido (Content-Based Image Retrieval, CBIR) tiene la ventaja de que al usar técnicas de análisis del propio contenido visual de la imagen, su manejo es bastante sencillo. El gran inconveniente de CBIR, se encuentra en que no permiten realizar consultas a un nivel conceptual elevado, puesto que las descripciones carecen de semántica.

Pese a las posibles desventajas, la recuperación de imágenes es muy necesaria debido al ritmo exponencial con el que crece la información digital y que se pone cada día a disposición del público con tecnologías como Internet o similares. Se necesitan métodos automáticos para la clasificación y el etiquetado de imágenes, que evitarían a los seres humanos el lento proceso de describir las imágenes manualmente y que además crearían unos conceptos mínimos para una correcta clasificación.

1.2 OBJETIVOS

El objetivo de este proyecto es realizar el diseño y la implementación de un sistema automático de detección y etiquetado de caras humanas.

El objetivo principal es, dada una fotografía en la que pueden aparecer un número variado de personas, detectar el número de caras existentes, si las hay, enmarcar dichos rostros y etiquetar las caras en base a las seis etiquetas propuestas, que son: raza, color de ojos, color de pelo, barba, longitud de pelo y sexo. Aunque en este proyecto se usarán estas seis etiquetas, el sistema permite incluir tantos clasificadores como sea necesario para etiquetar todo tipo de características, por ejemplo, pendientes, tatuajes, etc.

Únicamente se debe tener en cuenta la información existente en dicha imagen, no pudiendo utilizarse ninguna anotación relativa a la misma.

Para alcanzar el fin de este proyecto se van a estudiar e implementar los algoritmos necesarios para la detección y el etiquetado de caras, siendo estas dos las subtarefas en las que se puede dividir el proceso. Asimismo, al final del desarrollo, se realizará una exhaustiva evaluación del sistema.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

La primera fase trata de la detección de caras, es decir, dada una imagen determinar si en ella aparecen subimágenes que representan caras humanas y localizarlas para su posterior tratamiento. Se implementan dos métodos de distinta complejidad para realizar este proceso, los cuales se compararán en el capítulo 4, Evaluación.

Si se detecta alguna cara, se pasa a la fase de etiquetado, donde se extraen las características pertenecientes a dicha cara y a través de ella, se clasifican en torno a unas etiquetas.

En la presente memoria se estudiarán y analizarán cada una de estas etapas y se realizará un examen detallado de las mismas incluyendo un conjunto de pruebas experimentales que permitirán evaluar su comportamiento.

1.3 ESTRUCTURA DE LA MEMORIA

El contenido de la presente memoria está dividido en los siguientes capítulos.

- **Capítulo 1: Introducción**

Introducción general, presentación de los objetivos y descripción de la organización de la memoria.

- **Capítulo 2: Estado del arte**

Descripción de las etapas de Análisis de Imágenes y de las técnicas existentes sobre detección de caras. Además introduce información sobre los foros de Recuperación de Imagen, ImageCLEF y Pascal VOC.

- **Capítulo 3: Diseño e Implementación**

Arquitectura del sistema, técnicas de preprocesado, técnicas de segmentación, extracción de características y etiquetado.

- **Capítulo 4: Resultados**

Muestra los resultados experimentales obtenidos en la evaluación del sistema.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- ***Capítulo 5: Conclusiones y Trabajos Futuros***

Revisión de los objetivos propuestos y posibles trabajos futuros como continuación a este Proyecto Fin de Carrera.

- ***Anexo***

Tablas con los resultados de la evaluación de los algoritmos, manual de usuario y el código desarrollado en MATLAB.

- ***Bibliografía***

Referencias bibliográficas utilizadas en este documento.

2. ESTADO DEL ARTE

En este capítulo se van a introducir una serie de conceptos generales sobre la imagen digital y su procesamiento. Posteriormente se expondrá una descripción de las diferentes técnicas para la detección y el etiquetado de imágenes.

2.1 CONCEPTOS GENERALES

El análisis de imágenes es un proceso que consiste en descubrir, identificar y comprender los patrones relevantes en las imágenes objeto de estudio. En la figura 2.1 se muestran las etapas típicas de este proceso [11].

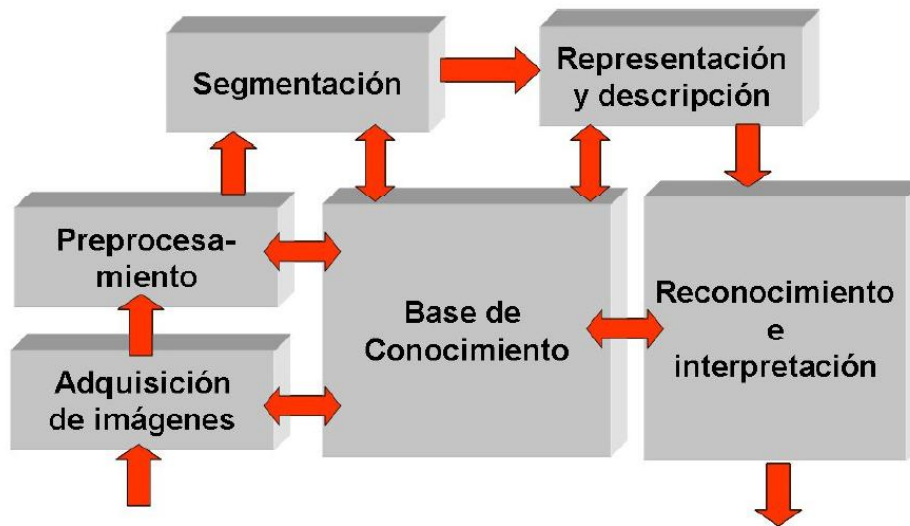


Figura 2.1: Diagrama de bloques del análisis de imágenes

La primera etapa del proceso es la *adquisición de la imagen*. Esta puede ser analógica, a la cual aplicaremos una digitalización (muestreo y cuantificación) o digital.

Una vez se ha obtenido la imagen digital, se pasa al *preprocesado*, que implica la mejora de la imagen de forma que se aumenten las posibilidades de éxito en las etapas posteriores.

La *segmentación*, consiste en agrupar píxeles, conforme a algún criterio de homogeneidad, y dividir la imagen en regiones de interés, llamadas objetos.

Una vez tenemos los objetos se trata de *representar* esa imagen mediante la creación de esquemas que compacten los datos y faciliten la labor de los descriptores.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

La *descripción* o selección de rasgos consiste en extraer rasgos o características dotadas de información cuantitativa de interés capaz de diferenciar una clase de objetos de otra.

La última etapa incluye el reconocimiento y la interpretación. Entendemos por *reconocimiento* el proceso que asigna una etiqueta a un objeto basándose en la información proporcionada por sus descriptores y la *interpretación* es aquella que le da un significado al objeto reconocido.

Tras este breve repaso, a continuación se explicarán algunas de estas etapas, en mayor o menor extensión, en función de la importancia que tengan para este proyecto.

2.1.1 LA IMAGEN DIGITAL

El píxel es la abreviatura de las palabras inglesas "picture element" y es el menor de los elementos de una imagen al que se puede aplicar individualmente un color o una intensidad. En el modelo matemático de una imagen, un pixel se identifica por su centro, de esta forma pueden representarse los pixeles como puntos (x,y) del plano.

Una imagen digital está compuesta por una agrupación de pixeles, cada uno con un valor de intensidad o brillo asociado. Una imagen digital se representa mediante una matriz bidimensional, de forma que cada elemento de la matriz se corresponde con cada pixel en la imagen.



a)

40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	40	40	40	40	40
40	40	40	40	200	200	40	40	40	40
40	40	40	40	200	200	40	40	40	40
40	40	200	200	200	200	200	200	40	40
40	40	200	200	200	200	200	200	40	40
40	40	40	40	200	200	40	40	40	40
40	40	40	40	200	200	40	40	40	40
40	40	40	40	40	40	40	40	40	40
40	40	40	40	40	40	40	40	40	40

b)

Figura 2.2: a) Imagen original; b) Estructura matricial de la imagen

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Dependiendo del rango de valores que pueda tomar cada pixel podemos distinguir los siguientes tipos de imágenes:

1. Imágenes binarias: el rango está formado únicamente por valores de negro y blanco $[0, 1]$.
2. Imágenes en escala de gris: existen hasta 256 niveles de gris, por lo que su rango se encuentra entre $[0, 255]$.
3. Imágenes en color: sus pixeles están cuantificados usando tres componentes independientes, por lo que todo color se puede componer a partir de esas tres componentes básicas.



Figura 2.3: a) Imagen en color; b) Imagen en escala de gris; c) Imagen binaria.

2.1.2 ESPACIOS DE COLOR

Las tres componentes que forman una imagen de color siguen un sistema de coordenadas tridimensional en el que cada color queda representado por un punto. Estos sistemas de coordenadas representan los distintos espacios de color.

La mayoría de los espacios de color empleados en la actualidad están orientados, bien hacia el hardware o bien hacia aplicaciones donde se pretende manipular el color. Los modelos orientados hacia el hardware utilizados, habitualmente son el RGB para monitores en color y una amplia categoría de cámaras de video de color, el CMY para impresoras en color y el YIQ para las televisiones a color. Entre los modelos que se emplean para las imágenes a color encontramos el HSI y el HSV [5, 25].

En nuestro caso nos centraremos en los espacios de color de RGB, YCbCr y HSV ya que son los empleados durante la implementación del proyecto.

2.1.2.1 Espacio de color RGB

El modelo de color RGB (1931), viene del inglés de Red-Green-Blue y está basado en la síntesis aditiva de las intensidades de luz relativas al rojo, al verde y al azul para conseguir los distintos colores [8].

La CIE (Comisión Internacional sobre Iluminación) definió las longitudes de onda de los colores primarios de luz como 700nm para el rojo, 546.1nm para el verde y 435.8nm para el azul.

En la figura 2.4 se aprecia la representación gráfica del espacio de color RGB que se realiza mediante un cubo unitario con los ejes R, G y B.

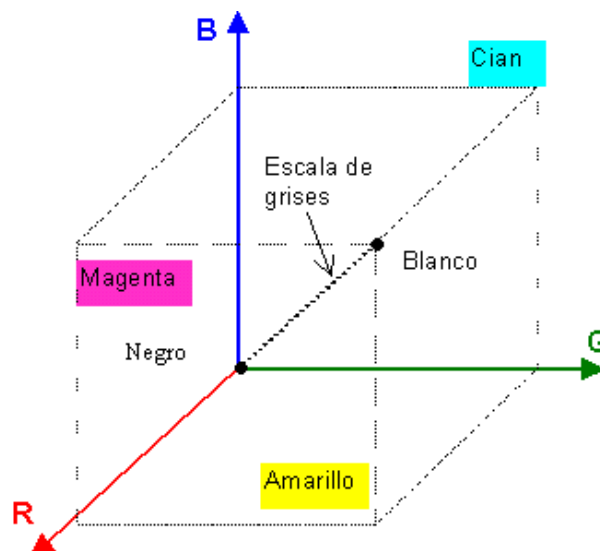


Figura 2.4: Cubo de coordenadas RGB

En este sistema los colores vienen definidos por un punto del cubo, así los vértices comunes son los colores secundarios, el origen de coordenadas es el negro (0 0 0), y el punto donde se sumarían los tres colores primarios será el vértice del color blanco (1 1 1). En la línea que une el negro con el blanco se representa toda la escala de grises.

Este modelo es el más utilizado para visualizar imágenes digitales en una pantalla en los formatos actuales por lo que tiene una gran importancia en el procesamiento de imágenes.

2.1.2.2 Espacio de color YCbCr

Uno de los conceptos importantes introducidos en el procesamiento de imagen, fue la separación de la luminancia de la crominancia. El modelo YCbCr es una codificación no lineal del espacio RGB, que representa el color como la luminancia (Y) y dos valores diferentes de color (Cb y Cr).

El parámetro Y indica la luminancia o claridad del color (que se pueden ver como un tono de gris) y los parámetros Cb y Cr ubican el color en una escala entre el azul y el amarillo (Cb) y en una escala entre el rojo y el verde (Cr).

La obtención de este espacio de color a partir del RGB, siendo R, G y B los valores de los canales rojo, verde y azul respectivamente, es la siguiente [8].

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cr = R - Y$$

$$Cb = B - Y$$

La sencillez de la transformación y la separación explícita de las componentes de luminancia y de crominancia, hacen a este modelo un método atractivo para la modelización del color de la piel.

2.1.2.3 Espacio de color HSV

Sus componentes principales son el tono (Hue), la saturación (Saturation) y el valor (Value). El sistema de coordenadas que sigue es cilíndrico y el subconjunto del espacio en que se define el color es una pirámide de base hexagonal [8].

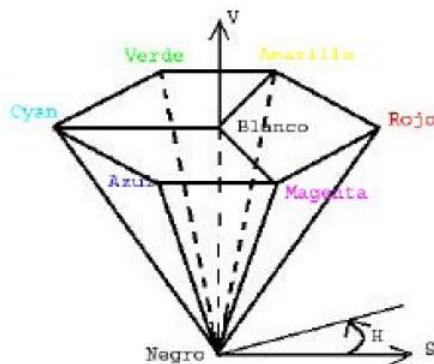


Figure 2.5: Cilindro de coordenadas de HSV

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

La componente H $[0^\circ, 360^\circ]$ describe el color en forma de ángulo. Los 0° representan el rojo, los 60° el amarillo, los 120° el verde, los 240° el azul y los 300° el magenta.

La componente S $[0, 1]$ indica cuánto de contaminado está el color o la cantidad de gris en proporción al tono. Si un color se desatura se convierte en un tono gris y si se satura completamente se obtiene el color puro.

La componente V $[0, 1]$ indica la claridad u oscuridad del color, es decir, la cantidad de luz del color.

La obtención de este espacio de color a partir del RGB, siendo R , G y B los valores de los canales rojo, verde y azul, es la siguiente [8].

$$H = \arccos \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{((R - G)^2 + (R - B)(G - B))}}$$
$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B}$$
$$V = \frac{1}{3}(R + G + B)$$

2.1.3 PROCESAMIENTO DE LA IMAGEN

Las técnicas de preprocesado pretenden mejorar o realzar las propiedades de la imagen para facilitar las etapas que le siguen en el análisis de imágenes. Los métodos de mejora se pueden basar en operaciones puntuales u operaciones de vecindad (7).

2.1.3.1 Operaciones puntuales

Las operaciones puntuales son aquellas que no dependen de la vecindad, es decir son independientes de la posición del pixel.

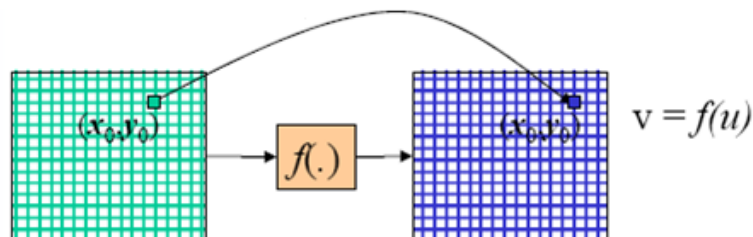


Figura 2.6: Operación puntual de un pixel

2.1.3.1.1 Modificación del contraste

Con la manipulación del contraste se pretende favorecer unas zonas en perjuicio de otras y distanciar los valores de los píxeles [7].

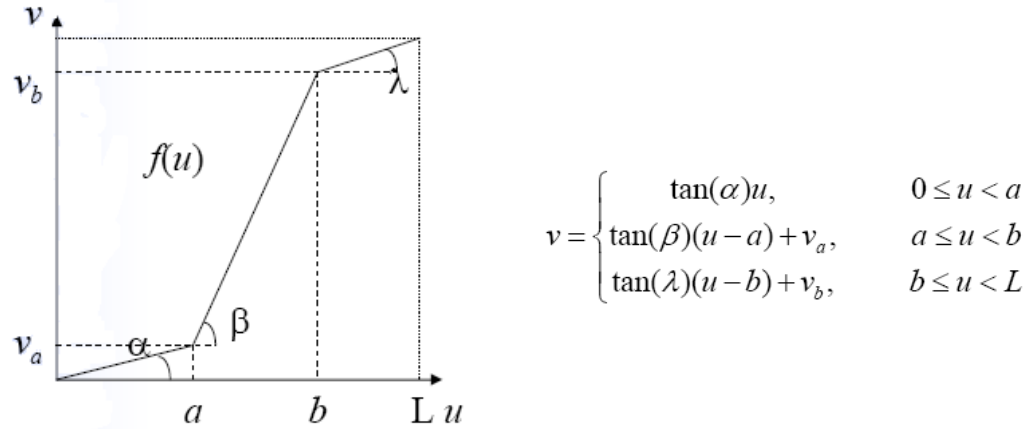


Figura 2.7: Funciones para la modificación del contraste.

Como vemos en la figura 2.7, tenemos una gráfica que nos representa distintos procedimientos de manipulación de contraste. En función de los parámetros β , α y λ , obtendremos unas modificaciones u otras.

Recorte: si $\alpha=\lambda=0$, estamos recortando la imagen para todos los valores menores que a y todos los valores mayores que b , mientras que los centrales quedarían afectados por la función correspondiente.

Umbralización: si $\alpha=\lambda=0$ y $a=b$, se mantendrán intactos los valores superiores a b , mientras que el resto se pondrán a cero. Podemos ver un ejemplo de umbralización en la figura 2.8.

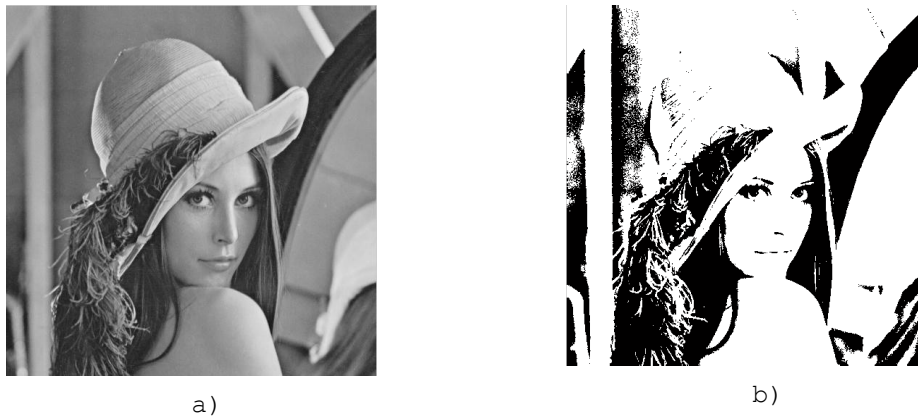


Figura 2.8: a) Imagen original; b) Umbralización siendo $a=b=100$

2.1.3.1.2 Modelado del histograma

El histograma es la representación de la frecuencia relativa de cada color en una imagen. Por ello la modificación del histograma trata de actuar sobre el conjunto de la imagen de forma global, con el objetivo de que el histograma se ajuste a una forma determinada.

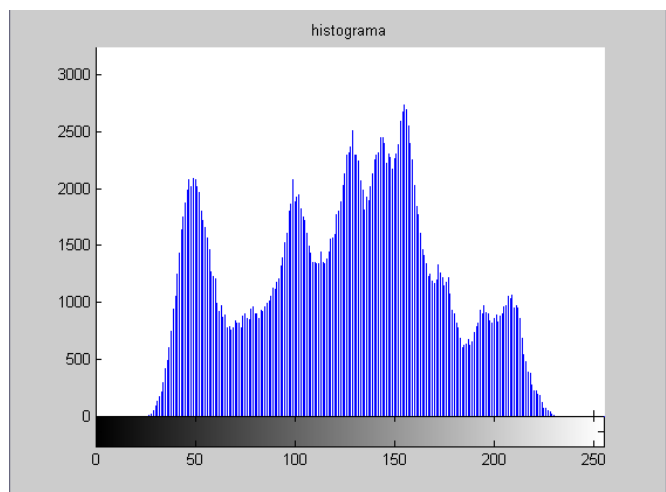
La *ecualización del histograma* pretende que para todos los niveles de gris se tenga el mismo número de píxeles, es decir, que sea uniforme (ver Figura 2.9).

La *especificación del histograma* sirve para obtener un histograma concreto en la imagen de salida.

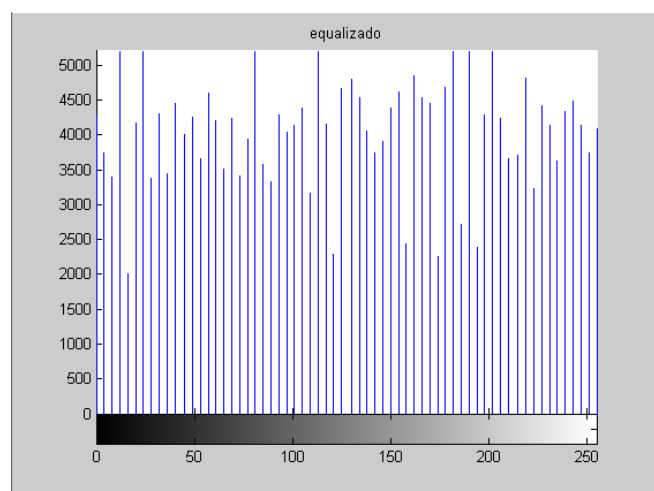
Las operaciones basadas en el histograma pueden efectuarse a partir de histogramas locales: de este modo, el perfil del histograma se adapta a las propiedades locales de la imagen [7].



a)



b)



c)

Figura 2.9: a) Imagen original; b) Histograma de la imagen; c) Histograma ecualizado de la imagen.

2.1.3.1.3 Operaciones entre imágenes

Son extensiones directas de las operaciones punto a punto y algunas de ellas tienen mucha importancia en el procesamiento de las imágenes [7].

La *suma* nos permite superponer imágenes y hacer un promediado para reducir el ruido aleatorio aditivo.

La *resta* elimina interferencia aditiva y sirve también para detectar el movimiento entre imágenes de la misma escena.

La multiplicación permite la eliminación de partes de una imagen si el producto se realiza con una máscara, de manera que solo se conserva lo que está debajo de ella.

El *máximo* y *mínimo* son operadores no lineales que nos permiten combinar imágenes (superposición).

2.1.3.2 Operaciones espaciales

Las operaciones espaciales o de vecindad, son aquellas operaciones "locales" en las que se modifica el valor de cada pixel de acuerdo con los valores de los pixeles que lo rodean. Se definen en un entorno " E_n " (vecindad) del punto a transformar (m_o, n_o) [7].

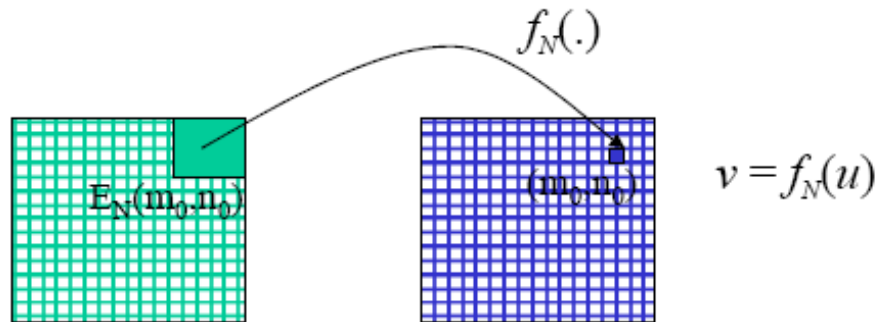


Figura 2.10: Operación espacial de un pixel.

La frecuencia espacial define la magnitud de cambios de los datos por unidad de distancia en una determinada zona de la imagen. Áreas de la imagen con pequeños cambios o con transiciones graduales en los valores de los datos se denominan áreas de bajas frecuencias. Áreas de grandes cambios o rápidas transiciones se conocen como áreas de altas frecuencias. Los filtros espaciales, se pueden dividir en tres categorías:

2.1.3.2.1 Filtro paso bajo

El espectro del ruido corresponde a componentes de alta frecuencia, por tanto, la aplicación de un filtro paso bajo significará la reducción del ruido sobre la imagen. El precio a pagar será una pérdida en la calidad de los detalles finos y en la degradación de los bordes.

Para la aplicación de los filtros se utilizará la relación fundamental de los sistemas discretos. No es necesario actuar sobre el espectro de la propia imagen para filtrarla frecuentemente, solo se requiere aplicar una máscara de convolución cuya respuesta frecuencial corresponda a la cancelación del espectro deseado (22). Todos los coeficientes de la máscara deben ser positivos y la suma de ellos debe ser igual a uno.

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Figura 2.11: Máscara de convolución de 3x3

Por ejemplo, en la figura anterior se realiza una media aritmética de los nueve píxeles que componen la ventana de filtrado, con lo que se reducen los espurios y la variabilidad de la imagen.

Como el filtro paso bajo difumina los bordes y emborrona la imagen, se puede usar otro filtro paso bajo que da mejores resultados. Este es el filtro de medianas, el cual en vez de usar la media usa la mediana que es menos sensible a valores extremadamente desviados.

2.1.3.2.2 Filtro gaussiano

Estos filtros se implementan para la eliminación del ruido gaussiano presente en las imágenes. Este ruido tiene un efecto general en toda la imagen, es decir, la intensidad de cada pixel de la imagen se ve alterada en cierta medida con respecto a la intensidad en la imagen original [6].

La función de transferencia del filtro gaussiano es la siguiente:

$$h_{\sigma}(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

donde σ es la desviación típica y σ^2 es la varianza.



Figura 2.12: a) Imagen con ruido $N(0, 0.03)$; b) Imagen filtrada con un filtro gaussiano de 3×3

2.1.3.2.3 Filtro paso alto

Enfatizan las altas frecuencias, eliminando las bajas, para mejorar o afilar las características lineales. Permitir el paso de las altas frecuencias produce como resultado una imagen llena de bordes y discontinuidades en la que la eliminación de los términos de baja frecuencia produce una reducción significativa del contraste global de la imagen [29].

Debe tener coeficientes negativos en la periferia y positivos en el centro, y la suma de todos ellos debe ser cero. Así, cuando la máscara se encuentre sobre una zona uniforme, la salida proporcionada por la máscara será cero o próxima a dicho valor.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Figura 2.13: Máscara de convolución de 3×3 Kernel.

2.1.3.2.4 Filtros detectores de bordes

Se entiende como borde aquella región donde aparece una fuerte variación del nivel de intensidad en los píxeles vecinos. Sin embargo, estas discontinuidades también aparecen de forma no deseada por la presencia del ruido, lo que hace más complicada la tarea de localizar los bordes [22].

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Las técnicas más usadas para detectar bordes son: los operadores de gradiente, los operadores laplacianos y el filtro de Canny.

1. Operadores de gradiente: en el dominio continuo, estos operadores constituyen la derivada primera en las direcciones x y y , mientras que en el dominio discreto son aproximaciones a ecuaciones diferenciales de primer orden, normalmente implementadas con operadores de convolución. Existen diferentes máscaras de convolución, por ejemplo para un entorno 2×2 se usa el operador Roberts, mientras que para uno de 3×3 se puede usar el Prewitt o el Sobel [29].
2. Operadores laplacianos: en el dominio continuo constituyen la derivada segunda en las direcciones x y y , es decir los pixeles que sean paso por cero. La laplaciana vale cero si $f(x,y)$ es constante y cambia linealmente su amplitud ante la existencia de un borde, por este motivo detectará con mayor precisión los bordes que los operadores de gradiente.
3. Filtro de Canny: se caracteriza por evitar la ruptura de los bordes de los objetos. Pretende maximizar la relación señal a ruido de la imagen, disminuir todo lo posible la distancia entre el borde detectado y el borde real y no identificar un borde por un único pixel, sino por un conjunto de pixeles que tengan una cierta conectividad.

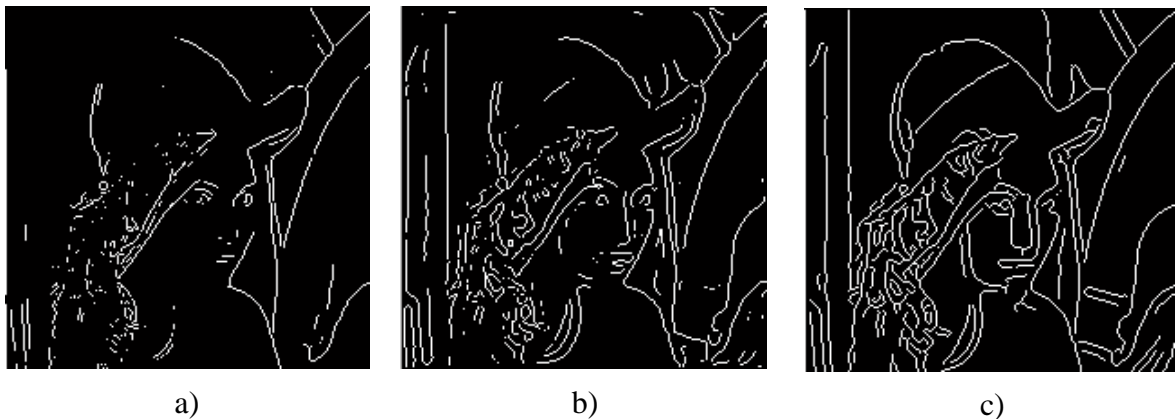


Figura 2.14: Imagen 2.9 a) filtrada por: a) Sobel; b) Laplaciano; c) Canny.

2.1.3.3 Filtrado en frecuencia

En el dominio frecuencial también puede realizarse el proceso de filtrado, con mayor grado de compresión de lo que se está viendo, ya que en una imagen en el dominio frecuencial se sabe donde se encuentran los distintos rangos de frecuencias. De esta forma, en vez de realizar la convolución, se efectúa su operación correspondiente en el dominio de la frecuencia: el producto (26).

$$G(u, v) = H(u, v) * F(u, v)$$

2.1.3.3.1 Filtros morfológicos

La morfología matemática es un método no lineal de procesar imágenes digitales basando en su forma. Su principal objetivo es la cuantificación de estructuras geométricas, Aquí los filtros también vienen definidos por su kernel, pero no es un kernel de convolución sino un elemento estructurante (26).

1. Dilatación

Su objetivo es rellenar pequeños agujeros y bahías de tamaño igual o menor que el elemento estructurante.

$$\delta_B(X) = X \oplus B = \{x | X \cap B_x \neq \emptyset\}$$

El resultado de la dilatación es el conjunto de puntos barridos por el centro del elemento estructurante, EE, mientras algún punto de B coincide con alguno de la imagen X. Es una operación extensiva, es decir el resultado contiene a la imagen original [7, 23].



Figura 2.15: a) Imagen binaria; b) Imagen dilatada con un EE en forma de disco y de radio 3.

2. Erosión

Su objetivo es eliminar grupos de píxeles donde no entra el elemento estructurante, como islas pequeñas o protuberancias.

$$\varepsilon_B(X) = X \ominus B = \{x \mid B_x \subseteq X\}$$

La salida de la erosión es el conjunto de puntos barridos por el centro del elemento estructurante mientras se cumpla que todos los puntos de B están contenido en X. Es una operación antiextensiva, es decir la imagen contiene al resultado [7, 23].



Figura 2.16: a) Imagen binaria; b) Imagen erosionada con un EE en forma de disco y de radio

3. Apertura y Cierre

La apertura es una combinación de erosión más dilatación, y se considera una operación antiextensiva e idempotente (si se realiza dos veces, queda la imagen original).

$$\gamma_B(X) = X \circ B = \delta_B(\varepsilon_B(X))$$

Alguno de sus efectos son: alisa contornos, elimina protuberancias, separa objetos en puntos estrechos y suaviza los bordes. Vemos estos efectos en la Figura 2.17.

El cierre es una combinación de dilatación más erosión y se considera extensivo e idempotente [7, 23].

$$\varphi_B(X) = X \bullet B = \varepsilon_B(\delta_B(X))$$

Algunos de sus efectos son: rellena vacíos y grietas, elimina entrantes y conecta objetos vecinos, como puede comprobarse en la Figura 2.17.

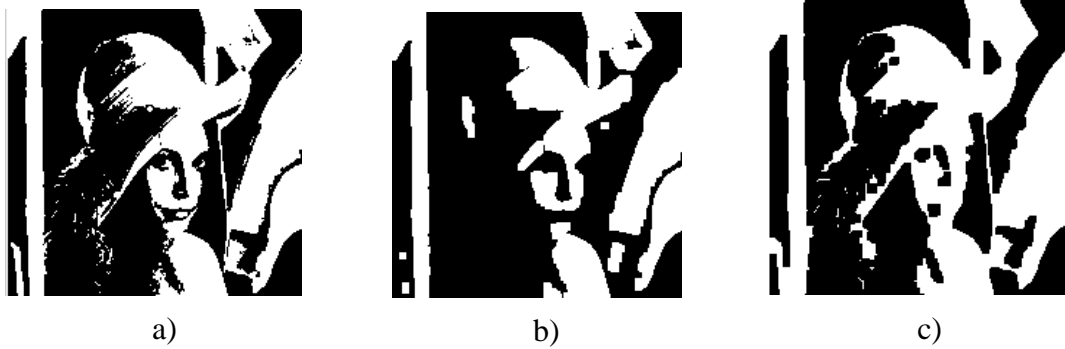


Figura 2.17: a) Imagen binaria; b) Imagen tras la apertura; c) Imagen tras el cierre.

2.1.3.3.2 Filtros de textura

Muchas imágenes contienen regiones caracterizadas por variaciones del nivel de gris, más que por un valor único de grises. La "textura" se refiere precisamente a la variación espacial del nivel de gris de una imagen como función de escala espacial [26].

1. Filtro de recorrido o de "rango": sustituye el valor central de la ventana de procesamiento por la diferencia entre el valor máximo y mínimo (el recorrido, estadísticamente hablando) de los píxeles contenidos en esa ventana. El recorrido será un valor pequeño para zonas "planas" o texturalmente uniformes, y será alto en zonas de alta variabilidad. El tamaño de la ventana debe ser suficientemente grande como para incluir un número suficiente de puntos. Según la escala a la que queramos trabajar, esto significa que debe ser mayor que el tamaño de cualquier pequeño detalle que pueda estar presente. El resultado es una imagen donde el valor de cada punto representa la textura y diferentes regiones pueden estar distinguidas por diferentes niveles de gris.
2. Filtro RMS (Root-Mean-Square): este filtro calcula primero la varianza de los valores de la ventana, y sustituye el valor centro por el RMS de los píxeles de la ventana de proceso.
3. Operadores de momento: el primer y el segundo momento son simples medidas de textura, utilizando los "momentos" del histograma de la ventana de proceso. El primer momento es una medida del contraste de la ventana. El segundo, es una medida de homogeneidad de la misma. Las imágenes resultantes pueden ser escaladas para crear una imagen que discrimina entre varias texturas.

2.1.4 SEGMENTACIÓN

La segmentación consiste en la división de la imagen en regiones que comparten ciertas propiedades. A estas regiones de interés se les llama objetos y el resultado de la segmentación es la separación de objetos [7].

La siguiente exposición versa sobre las distintas técnicas de segmentación clasificadas en:

- **Contorno de los objetos:** trata de localizar las fronteras de los objetos. Hace referencia a los aspectos geométricos de los objetos. En este contexto se verán las técnicas basadas en los bordes.
- **Búsqueda de regiones homogéneas:** agrupa los píxeles porque cumplen algún criterio de similitud y tienen conectividad entre ellos. En este contexto se verán la umbralización y clustering [22].

2.1.4.1 Técnicas basadas en los bordes

Para la localización de contornos en la escena se empleará, previo uso de las técnicas de segmentación, técnicas de detección de bordes [22]. Por lo tanto, será necesaria la implementación de una tarea adicional en la etapa de preprocesado de la imagen. Sin embargo, estas técnicas de segmentación no son definitivas en la búsqueda de objetos en la imagen. La presencia de ruido, el efecto de las sombras, la falta de iluminación uniforme y un largo repertorio de causas, produce que los contornos no sean del todo continuos y cerrados sobre los objetos. Se requiere, además de una etapa de preprocesado, una de post-procesado.

En esta línea muchas aplicaciones se basan en la localización de líneas rectas u otras formas básicas como círculo o elipses. En este contexto se aplica la **Transformada de Hough** para la localización de fronteras.

La entrada del algoritmo de Hough es una imagen binaria o en escala de grises, en la que se ha hecho detección de bordes. A partir de ella, se hace una transformación de dominio para encontrar curvas perimétricas.

Se obtiene un algoritmo robusto ante la presencia de ruido, pero a cambio de un alto coste computacional.

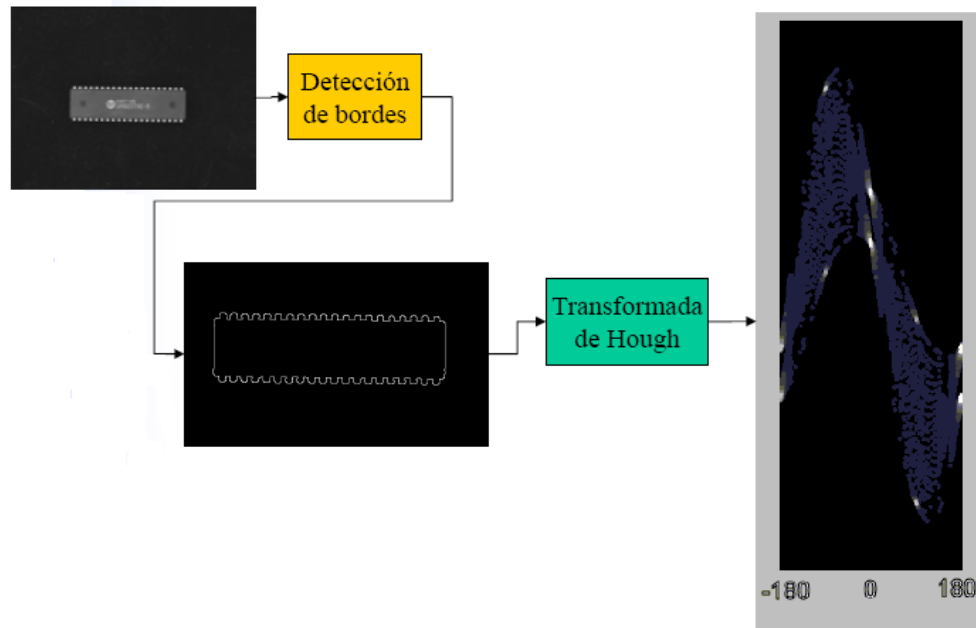


Figura 2.17: Ejemplo de Transformada de Hough.

2.1.4.2 Umbralización

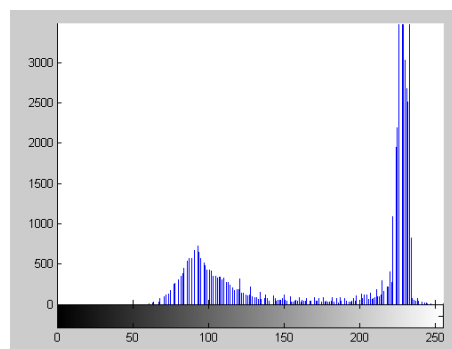
Se emplea cuando, en la escena, existe una clara diferencia entre los objetos y el fondo. Esta técnica se usa tras aplicar un umbral, con el cual la imagen en niveles de intensidad, se transforma en binaria.

El problema radica en la obtención del umbral, operación nada sencilla, ya que las imágenes están contaminada de ruido. Con el fin acotar la memoria, únicamente se explica una de las técnicas más utilizadas.

La **segmentación por histograma** [22] es una técnica global, es decir el umbral solo depende del nivel de intensidad, que considera una iluminación relativamente uniforme en la escena y una definición clara de los objetos respecto al fondo.



a)



b)

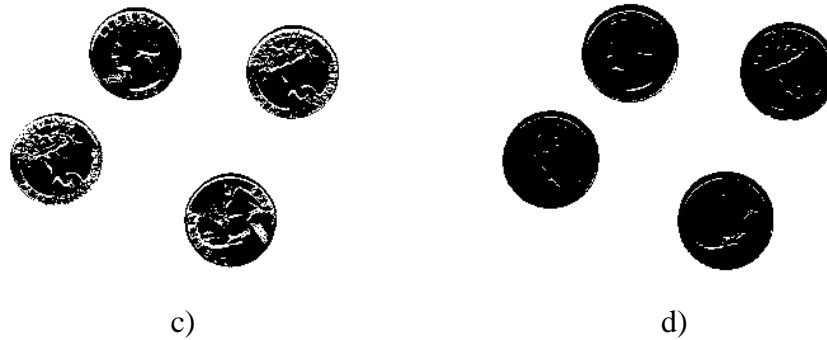


Figura 2.18: a) Imagen en escala de gris; b) Histograma; c) Imagen binaria resultante de aplicar umbral=120; d) Imagen binaria resultante de aplicar umbral=180.

Se trata de crear el histograma de una imagen, donde se obtendrán dos picos y un valle (a mayor número de picos y valles, más umbrales y por lo tanto, mayor complejidad) y situar el umbral en el valle.

Un resumen detallado de los distintos algoritmos para la selección óptima del umbral se puede encontrar en revistas especializadas [23]. De entre todas las propuestas se va a destacar el umbral de Otsu, por su generalizada utilización en las aplicaciones de Visión Artificial.

2.1.4.3 Técnicas de agrupamiento o clustering

Este tipo de segmentación se fundamenta en dos principios: la definición de un patrón o centroide por cada región del espacio de características, y la asignación de cada vector de características al centroide más próximo [7].

Esta forma de proceder es útil cuando no se conocen ni las características de las regiones a buscar ni cuantas categorías hay.

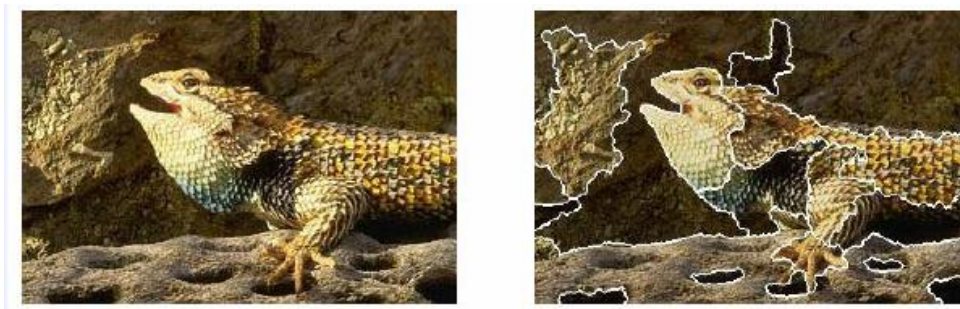


Figura 2.19: Ejemplo de clustering.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

El agrupamiento puede realizarse mediante el algoritmo de las k-medias, que trata de encontrar k prototipos a partir de los vectores de un conjunto dado. Se inicia el método con k muestras aleatorias, se calcula la distancia de cada dato a cada uno de los k representantes y se asigna a aquel con el que guarde mínima distancia. Finalmente, se recalculan los representantes de cada grupo o clase y vuelve a aplicarse de forma iterativa.

El algoritmo de las k-medias supone conocido el número de regiones de la imagen. Pero, puede ocurrir que esto sea desconocido, en este caso, se procede con la división de regiones. En esta técnica se parte de dos centroides (regiones), se aplica un algoritmo de agrupamiento (por ejemplo k-medias), se define un criterio de homogeneidad para determinar donde separar las regiones vecinas, y si no se satisface el criterio de parada, se dividen las regiones vecinas menos homogéneas y se vuelve a aplicar el algoritmo de agrupamiento; es un proceso iterativo.

Es por tanto de vital importancia, la selección de un buen criterio de parada, ya que con él se determinará correctamente el número de regiones.

2.1.5 REPRESENTACIÓN Y DESCRIPCIÓN

Después de segmentar una imagen, el conjunto resultante de píxeles se representa y describe de forma conveniente para su posterior reconocimiento. Con el fin de no desviarse de los propósitos de la memoria, se hará un breve resumen, para mayor información consultar [7].

2.1.5.1 Representación

Las técnicas de segmentación producen datos en bruto en forma de píxeles de un contorno o región. Aunque a veces es posible obtener los descriptores a partir de estos datos, en general, será necesaria la utilización de esquemas que compacten los datos en representaciones a partir de las cuales el cálculo de descriptores sea más útil. A continuación se describen alguno de los métodos existentes:

- **Códigos de cadena:** se utilizan para representar un contorno por medio de la sucesión de segmentos de longitud y dirección especificadas. Se recorre el contorno a partir de un punto de inicio prefijado en dirección de las agujas del reloj.

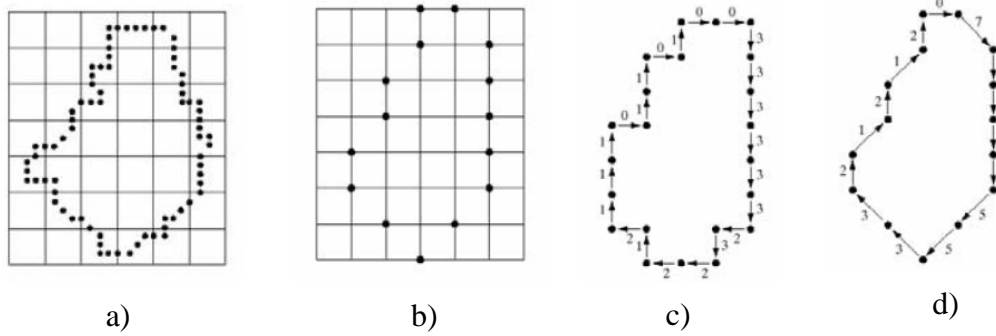


Figura 2.20: a) Contorno original; b) Remuestreo; c) Código de cadena con conectividad a 4; d) Código de cadena con conectividad a 8.

- **Aproximaciones poligonales:** buscan representar un contorno a través de un polígono. La aproximación únicamente es exacta cuando el número de lados del polígono es el mismo que el número de puntos del contorno. Sin embargo, en la práctica se busca captar la esencia del contorno a través del menor número posible de lados. Alguno de los métodos que se pueden emplear son: polígonos de perímetro mínimo, técnicas de fusión y técnicas de división, los cuales se especifican en [7].

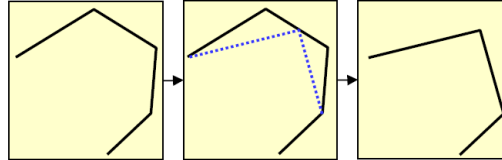


Figura 2.21: Ejemplo de técnica de fusión.

- **Signaturas:** representación de 1D de la frontera. Una de las firmas más simples se obtiene al representar la distancia del centro geométrico al contorno en función del ángulo, como se puede ver representado en la Figura 2.22.

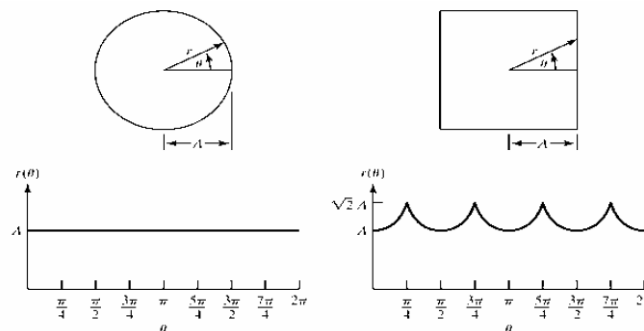


Figura 2.22: Ejemplos de firmas distancia-ángulo.

2.1.5.2 Descripción

La descripción está orientada a la extracción de características tanto útiles como significativas para la tarea de reconocimiento. Existen descriptores de forma, de región y relacionales [7].

2.1.5.2.1 Descriptores de forma

- **Descriptores simples:** algunos de los descriptores simples son longitud del contorno, diámetro del contorno y curvatura.
- **Números de forma:** se denomina así al número que forma el código de primera cadena. El orden es el número de dígitos de su representación.
- **Descriptores de Fourier:** se forma un número complejo con las coordenadas de los píxeles del borde y se le aplica su Transformada Discreta de Fourier (DFT) [9nieves]. Los descriptores corresponden al módulo de los coeficientes complejos.
- **Momentos estadísticos:** se trata el contorno como una variable aleatoria y se calculan sus momentos.

2.1.5.2.2 Descriptores de región

- **Descriptores simples:** son el área, perímetro, compacidad, excentricidad, etc.
- **Descriptores topológicos:** utilizan el número de componentes, conectadas, el número de agujeros y el número de Euler para describir la región.
- **Textura:** proporcionan información de suavidad, tosquedad y regularidad. Existen tres métodos: estadísticos, estructurales y espectrales.

2.1.5.2.3 Componentes principales, PCA

El objetivo es resumir un grupo de variables en un conjunto más pequeño, sin perder por ello una parte significativa de la información original. Se utiliza las medias de cada componente de color R, G, B, y la matriz de covarianzas. Finalmente, se obtienen los autovalores y autovectores, y el autovector asociado al mayor autovalor, será el que determine la componente principal [7].

2.1.6 RECONOCIMIENTO

Tras los procesos de segmentación, extracción de características y descripción, cada objeto queda representado por una colección (posiblemente ordenada y estructurada) de descriptores, denominada patrón. Se trata de clasificar dichos patrones según la pertenencia del objeto a una clase u otra [7, 23].

Se ha de seleccionar adecuadamente el patrón a utilizar, ya que el funcionamiento del reconocedor dependerá de que la información obtenida sea suficientemente discriminante. Los patrones que describen objetos de la misma clase deben presentar características similares mientras que patrones que describan objetos de diferentes clases deberán presentar características diferenciadas.

2.1.6.1 Métodos de decisión teórica

Estos métodos se basan en la utilización de funciones de decisión o funciones discriminantes [11].

2.1.6.1.1 Adaptación

La idea consiste en representar cada clase mediante un patrón prototipo. El patrón a clasificar se asigna a la clase más cercana en términos de una métrica de distancia predefinida. Esta representación puede hacerse de varias formas, se presentarán dos de ellas, mediante mínimas distancias y mediante correlaciones [7, 23].

- **Clasificador de mínima distancia:** la función de decisión viene definida por:

$$d_i(x) = \|x - m_i\|^2 \qquad m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}$$

Siendo m_i el representante de cada categoría. Por tanto, se calculan los representantes de cada categoría, y se asigna cada muestra a la clase "más cercana".

- **Adaptación por correlación:** consiste en la comparación de la imagen a clasificar con una o varias imágenes patrón que caracterizan a cada clase. La forma más simple es utilizando la correlación entre la imagen y una máscara.

2.1.6.1.2 Redes Neuronales

Las redes neuronales engloban un conjunto de métodos capaces de obtener directamente las funciones de decisión requeridas mediante entrenamiento [7, 23].

- **Perceptrón para dos clases de patrones:** En su forma más simple, el perceptrón aprende una función de decisión lineal que separa completamente, y en un determinado número de iteraciones, dos conjuntos de entrenamiento linealmente separables.

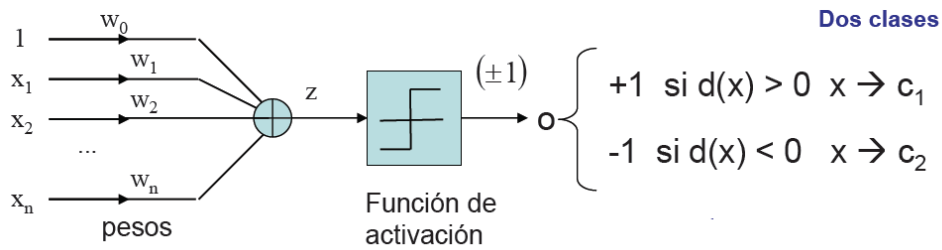


Figura 2.23: Perceptrón para dos clases de patrones.

- **Perceptrón multicapa:** la red multicapa permite obtener las funciones de decisión para los problemas de reconocimiento de patrones multiclase de forma independiente a la separabilidad de las clases. En la Figura 2.24 se ve una red neuronal con numerosas capas de nodos de cálculo o neuronas, estructuralmente idénticas, de forma que la salida de cada una de ellas alimenta las entradas de todas las de la siguiente capa. La red reconoce un patrón vectorial x como perteneciente a la clase c_m , si la m -ésima salida de la red está activada y todas las demás desactivadas.

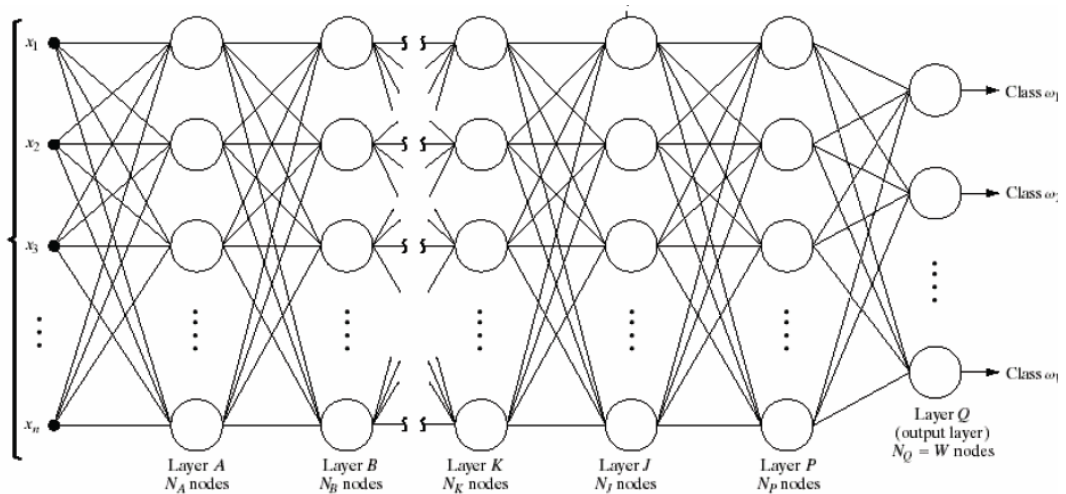


Figura 2.24: Perceptrón multicapa progresivo.

2.2 DETECCION DE CARAS

La primera fase del proyecto es el proceso de detección de caras, es decir, dada una imagen determinar si en ella aparecen subimágenes que representan caras humanas y localizarlas para su posterior tratamiento.

En la década de los setenta surgieron los primeros algoritmos de detección de caras, basados en técnicas heurísticas y antropométricas. Dichos algoritmos fallaban bastante, y eran muy sensibles a los cambios. En un principio solo detectaban una cara en una fotografía con fondo blanco.

Estas investigaciones se abandonaron debido a que por entonces la detección fácil no tenía utilidad, pero se volvieron a desarrollar en los años noventa (expresiones).

El problema de detectar caras en una imagen tiene diferentes aspectos problemáticos. Algunos de ellos son los siguientes [1]:

- ✓ Pose y orientación de la cara.
- ✓ Presencia de gafas, barba, gorro, etc.
- ✓ Expresión de la cara.
- ✓ Problemas de iluminación (no uniformidad).
- ✓ Condiciones generales de la imagen (ruido, fondo).
- ✓ Cantidad desconocida de caras en la imagen.

Además de la localización de caras en imágenes existen otros problemas relacionados con éste:

- ✓ Localización de características relevantes, como ojos, boca, cejas, orejas, etc.
- ✓ Reconocimiento de expresiones: triste, alegre, enfadado, etc.
- ✓ Estimación de la pose y seguimiento.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En la actualidad existe una gran cantidad de algoritmos de detección de caras con desempeño variado y dependiente de los escenarios a considerar. Para una descripción de alto nivel de los mismos, se pueden consultar los estudios de [32]. Una posible clasificación de los algoritmos de detección de rostros es:

- Técnicas basadas en rasgos faciales: Estos métodos se basan en buscar determinados elementos que componen una cara, como pueden ser los ojos, las líneas de contorno, etc.
- Técnicas basadas en la imagen: En este caso los métodos trabajan con la imagen completa o zonas concretas de la misma, efectuando cálculos que determinan si hay una cara o no.

A continuación se pasa a describir estos métodos en base a la Tabla 2.1.

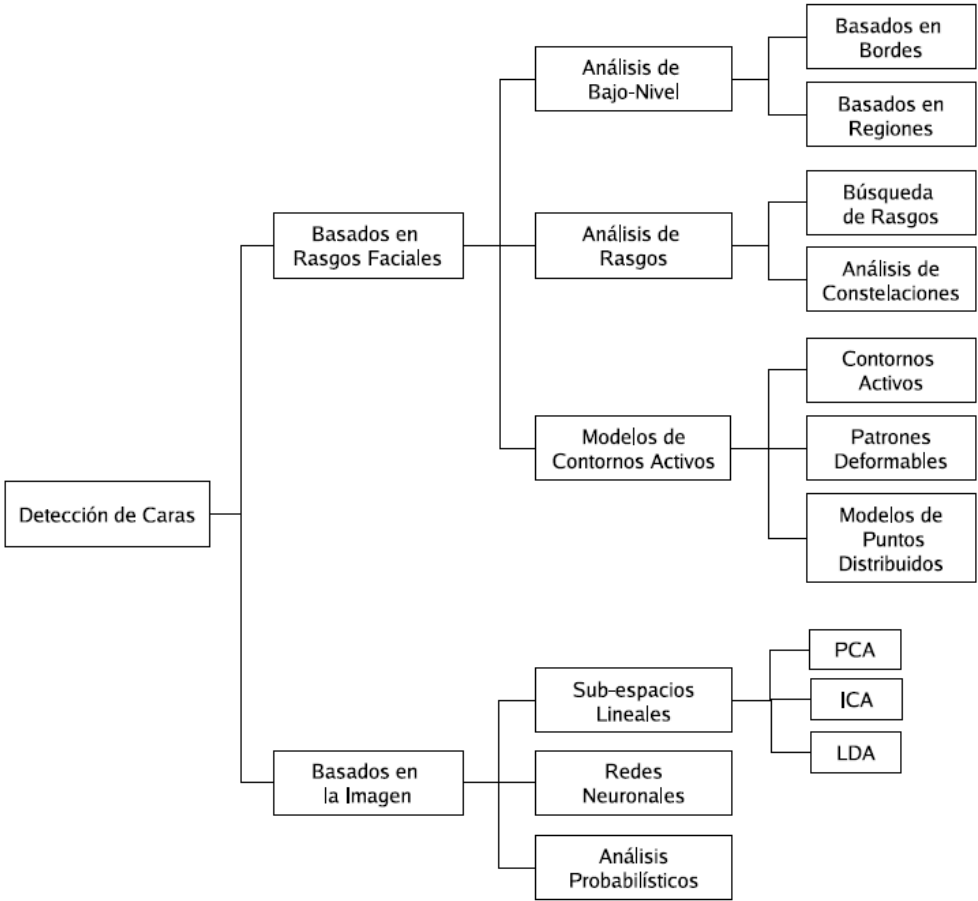


Tabla 2.1: Taxonomía de detección de caras

2.2.1. TÉCNICAS BASADAS EN RASGOS FACIALES

Estas técnicas explotan propiedades aparentes de la cara tal como el color de la piel y la geometría facial. La detección de la cara se resuelve manipulando medidas de distancias, ángulos y áreas de los rangos visuales en la imagen. Lo más importante en este tipo de métodos es decidir qué rasgos de la cara interesan para el estudio.

2.2.1.1. Análisis de bajo nivel

Son técnicas que trabajan a nivel de pixel. Hay diversos métodos dentro de este apartado, las más características son las siguientes:

2.2.1.1.1. Detección de bordes

La idea se basa en analizar las líneas que componen los bordes de una cara y utilizarlas para detectar los rasgos faciales. El algoritmo sigue los siguientes [2]:

- Detectar los bordes de la imagen.
- Una vez obtenidos los bordes, se efectúa un adelgazamiento a fin de obtener para cada borde una línea de un pixel de ancho que lo represente.
- Filtrado de componentes. El algoritmo se queda solo con las componentes que sean más susceptibles de formar parte de una cara. Por ejemplo, buscando líneas que en conjunto se asemejan a una elipse de determinadas proporciones de ancho y alto.
- Etiquetado. Una vez obtenidas dichas componentes, se etiquetan como lado derecho de la cara, lado izquierdo, línea de pelo, etc.

Las componentes etiquetadas se combinan para formar posibles candidatos para ser una cara, decisión que toma una función de coste, que utiliza la proporción áurea para sus cálculos.

Esta técnica detecta correctamente aproximadamente tres de cada cuatro caras, con un 2% de falsos positivos. Este algoritmo puede trabajar (aunque con más dificultad), incluso si el fondo de la imagen no es uniforme. Si la cara no está de frente, es totalmente ineficiente.

2.2.1.1.2. Información de grises

Trabaja sobre la idea de que rasgos faciales, tales como las cejas, las pupilas y los labios aparecen como zonas más oscuras de la imagen que las zonas que corresponden a las regiones faciales que los rodean. Este algoritmo se compone de las siguientes partes [2]:

- Aumenta el contraste de la imagen. De esta forma se resalta aún más la diferencia de luminosidad entre las citadas partes de la cara.
- Umbralización. El algoritmo se queda sólo con las zonas de la imagen cuyo valor de gris supere un cierto umbral.
- Detección de caras mediante el uso de plantillas ponderadas. Hasta aquí se ha obtenido una imagen compuesta por multitud de "manchas" negras. Este paso trata de comparar la distribución de esas manchas con las manchas "tipo" de una cara, usando plantillas.

Este algoritmo produce diferentes resultados en función del color de la piel del sujeto que aparece en la imagen. Para corregir este problema, existe una técnica similar aunque más potente que usa la información del color.

En este proyecto se van a utilizar técnicas de este tipo para la detección de cara, que se explicarán posteriormente en el capítulo de Diseño e Implementación.

2.2.1.2. Análisis de rasgos

El problema del análisis a bajo nivel es que se puede proporcionar información ambigua, por ejemplo, si aparecen en la imagen objetos que tenga un color similar al del modelo de color de piel utilizado.

El análisis de rasgos se basa en la geometría de la cara para caracterizar y posteriormente verificar rasgos a fin de evitar dicha ambigüedad.

2.2.1.2.1. Búsqueda de rasgos

Estas técnicas buscan rasgos prominentes que permiten localizar rasgos menos prominentes partiendo de hipótesis geométricas. Uno de los posibles algoritmos es el siguiente:

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- Búsqueda de la parte superior de la cabeza. Se efectúa una hipótesis sobre lo que puede ser una posible línea del pelo en lo alto de la frente. Puede ser difícil si la persona tiene el pelo cubriendo zonas de la frente.
- Búsqueda de los ojos. A partir de dicha línea efectúa un barrido hacia abajo tratando de buscar zonas donde la densidad de gris aumente y disminuya bruscamente en el plano horizontal. Dichas zonas corresponden con las pupilas. Falla si el individuo usa gafas, si uno de los ojos no aparece por cualquier motivo, o si la imagen está rotada
- Uso de plantillas flexibles. La distancia entre la línea del pelo y el plano de los ojos se usa como medida de referencia para inicializar una plantilla flexible que cubre el resto de rasgos, como la nariz y la boca. La plantilla trata entonces de ajustarse a dichos rasgos usando una función de costes basada en bordes.

2.2.1.2.2. Análisis de constelaciones

Se basan en el uso de un modelo probabilístico que estudia la posición espacial de los rasgos faciales, intentando buscar patrones que se asemejen a una cara.

2.2.1.3. Análisis de formas activas

Se basan en representar la imagen en rasgos de alto nivel para posteriormente interactuar con rasgos locales de la imagen (ojos, brillo) y gradualmente deformarla hasta adaptarla a la forma de los rasgos.

- Snakes: se usan comúnmente para localizar el contorno de la cara. Se basan en la minimización de una función de energía para adaptar el modelo. Se inicializa una snake ante una cara inminente, y posteriormente se va cerrando a los contornos de la imagen asumiendo la forma de la cara [1].
- Plantillas deformables: es el siguiente paso, usando las snakes para encontrar más rasgos faciales además del contorno de la cara. Por ejemplo se pueden encontrar los ojos usando para las snakes un mecanismo de deformación que incluye el cálculo de gradientes de una función de energía.

2.2.2. TÉCNICAS BASADAS EN LA IMAGEN

En estas técnicas, por el contrario, el objeto de estudio es la imagen misma. El conocimiento previo se incorpora implícitamente en esquemas de entrenamiento. Se trabaja directamente con una representación de la imagen a la que se le aplica algoritmos de entrenamiento y análisis.

2.2.2.1. Métodos basados en subespacios

Consideran las imágenes de caras humanas como un subespacio lineal de un espacio mayor (de todas las imágenes). La base del método es la siguiente [9]:

- Construir la base canónica. Partiendo de un conjunto de imágenes que representan caras (y sólo caras), se encuentran los componentes principales de una cara, expresados en términos de autovectores, aquí llamados eigenfaces. Cada cara del conjunto anterior puede ser aproximada por una combinación lineal de las eigenfaces usando los pesos apropiados.
- Detección de caras. Tomando la imagen global como punto de partida, se intentan representar todos los elementos de la imagen en el espacio de las eigenfaces. Cada fracción de la imagen representada produce un error residual llamada "distancia desde el espacio de las caras" (DFFS, distance from the face space). Observando los valores mínimos de estas distancias se puede determinar la posición de las caras en la imagen.

2.2.2.2. Redes neuronales

Es el método más utilizado en la detección de caras en una imagen, dado el alto porcentaje de acierto que produce, siendo en algunos casos superiores al 95%. Las redes neuronales se usan generalmente para la clasificación de imagen según patrones establecidos, por ejemplo, conseguir averiguar qué representa una imagen borrosa o incompleta. La red neuronal, como se explicó en 2.1.6.1.2, se entrena usando un conjunto de imágenes que representan caras de todo tipo (en cuanto a raza, tono de piel, posición de los labios, etc.) y otro conjunto de imágenes que no representan caras, de forma que la red neuronal pueda establecer el criterio adecuado acerca de lo que es una cara y lo que no lo es.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Cuando se trata de una imagen grande, compleja, en la que pueden aparecer varios rostros, se procede de la siguiente forma [9]:

1. Establecer el tamaño de cara mínimo reconocible en la imagen, medido en píxeles.
2. Crear una ventana cuadrada de la imagen del tamaño mínimo, partiendo de la esquina superior izquierda.
3. Introducir la porción obtenida de la imagen en la red neuronal, y determinar si hay cara o no.
4. Repetir el paso 3 para todas las posiciones posibles de la ventana de la imagen, desplazándola horizontal y verticalmente.
5. Repetir desde el paso 2, aumentando en cada iteración el tamaño de la ventana, hasta que la ventana alcance el tamaño de la imagen original.

Este algoritmo se caracteriza por ser más lento que los demás vistos hasta ahora, pero compensa por su alto porcentaje de aciertos.

2.2.2.3. Métodos estadísticos

Bastante similar al método anterior, también supone un entrenamiento del algoritmo. En este caso, se usa un método estadístico en vez de una red neuronal: lo que más se parezca estadísticamente a lo que yo sé que es una cara, tendrá más probabilidades de ser una cara.

El método concreto consiste en calcular la varianza entre dos funciones probabilísticas de densidad (creadas durante el entrenamiento), correspondientes a la probabilidad de que la imagen sea una cara, y a la probabilidad de que no lo sea. [9]

Aquí se incluyen métodos que utilizan máquinas de vectores de soporte y reglas de decisión de Bayes.

2.3 ETIQUETADO DE IMÁGENES

La segunda fase del proyecto, está basada en etiquetado de imágenes. Para esta parte sirve como guía los procedimientos que se están implementando a día de hoy en foros o congresos internacionales de procesamiento de imágenes como son CLEF ó PASCAL VOC.

2.3.1 CLEF

CLEF (Cross Language Evaluation Forum) es el foro europeo más importante para la evaluación de sistemas de recuperación multilingüe y multimedia. CLEF está financiado desde 2000 por la Unión Europea y está coordinado por el Istituto di Scienza e Tecnologie dell'Informazione, del Consiglio Nazionale delle Ricerche (CNR) en Italia. En él participan tanto grupos académicos como comerciales de todo el mundo [12, 21].

Rivaliza a nivel mundial con TREC (Text Retrieval Conference) en Norteamérica y NTCIR (NIIII Test Colecction for IR System) en Asia. A diferencia de TREC, que se centra en las técnicas de recuperación de información sobre el idioma inglés, y de NTCIR, centrado en los idiomas asiáticos como el japonés, chino y coreano, el objetivo de CLEF es la investigación en un gran número de idiomas (europeos y no europeos) y en las técnicas de recuperación cross-lingual, es decir, cuando el idioma de la consulta es diferente del de la colección de documentos en la que se busca. El objetivo final es promover y estimular el desarrollo de las tecnologías de recuperación de información en Europa para garantizar su competitividad en un ámbito mundial.

Típicamente las tareas que se crean en el foro, consisten en que los organizadores proporcionan a los participantes una vasta colección de objetos para indexar (noticias, páginas web, imágenes, etc.) y plantean una serie de objetivos de búsqueda (topics).

Así, los participantes construyen sus sistemas, ejecutan las búsquedas y envían sus resultados en un formato estandarizado. Por último, los organizadores realizan la evaluación y establecen un ranking de grupos en función de la precisión de su sistema. De esta forma, se pueden comparar las diferentes técnicas empleadas y analizar los resultados de forma colaborativa.

CLEF está dividido en diferentes áreas temáticas (tracks) que investigan diversos aspectos de la recuperación de información, como la búsqueda de texto (tare ad-hoc), la búsqueda de información geográfica (GeoCLEF), la búsqueda de información en la web (WebCLEF), la recuperación de imágenes (ImageCLEF), los sistemas de búsqueda de respuesta (QA@CLEF), etc. Luego cada track ofrece distintas tareas (tasks) sobre diferentes aspectos centrados en ese tema en concreto [21].

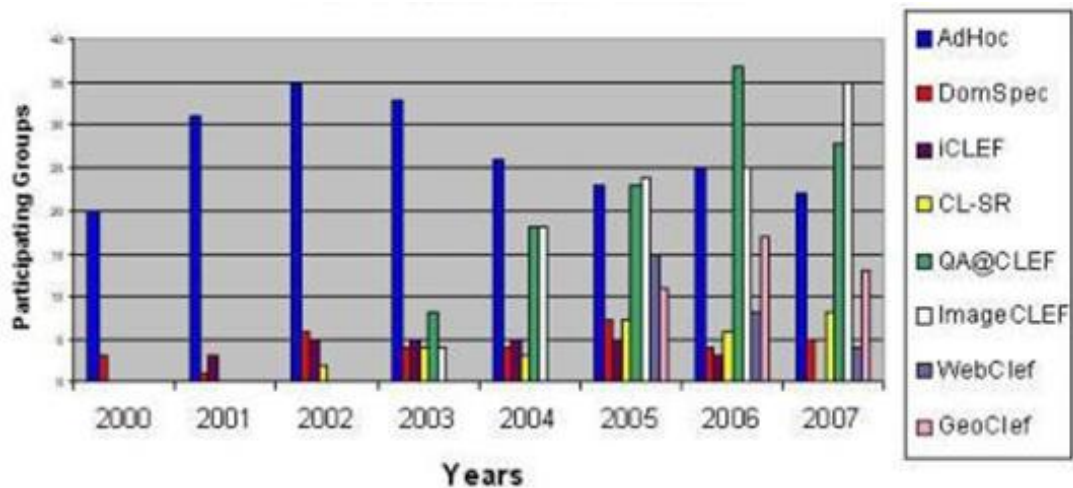


Tabla 2.2: Evolución desde 2000 al 2007 del número de participantes en cada área temática.

Para este proyecto el track que nos interesa es el de ImageCLEF, que como se ve, aparece en 2003 y va aumentando su participación de manera elevada con el paso del tiempo, hasta ser el de más popular en el año 2007.

ImageCLEF fue propuesto por Mark Sanderson y por Paul Clough de la Universidad de Sheffield, aunque hoy en día lo organizan de manera voluntaria un elevado número de investigadores internacionales.

Define varias tareas en las que los grupos participantes pueden concursar libremente sin límite de propuestas. Estas tareas se han ido modificando cada año desde que comenzó CLEF [12].

CLEF2003

- Recuperación automática ad hoc (Automatic ad hoc Retrieval).
- Recuperación interactiva de imágenes (Interactive Image Retrieval).

CLEF2004

- Recuperación de fotografías.
- Recuperación de imágenes médicas (Medical Image Retrieval).
- Búsqueda centrada en el usuario.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

CLEF2005

- Recuperación de fotografías.
- Búsqueda interactiva.
- Recuperación de imágenes médicas.
- Anotación automática de imágenes médicas (Automatic Annotation Task for Medical Images).

CLEF2006

- Recuperación de imágenes fotográficas.
- Recuperación visual ad hoc.
- Anotación automática de fotografías generales.
- Recuperación de imágenes médicas.
- Anotación automática de imágenes médicas.

CLEF2007

- **Recuperación de objetos (Object Retrieval Task).**
- Recuperación de imágenes fotográficas.
- Clasificación automática de imágenes médicas.
- Recuperación de Imágenes médicas.

CLEF2008

- **Recuperación de fotografías (ImageCLEFphoto).**
- Recuperación de imágenes médicas.
- **Tarea de detección de concepto de una fotografía.**
- Anotación automática de imágenes médicas.
- Recuperación de imágenes de una colección de imágenes de Wikipedia.

CLEF2009

- Tarea de recuperación de fotografías.
- **Anotación de fotografías utilizando ontología simple.**
- Recuperación de imágenes médicas.
- Tarea visual de una imagen robótica.
- Anotación automática de imágenes medicas.
- Recuperación de imágenes de una colección de imágenes de Wikipedia.

CLEF2010

- Recuperación de imágenes médicas.
- Recuperación de fotografías.
- Tarea visual de imagen robótica.
- Recuperación de imágenes de una colección de imágenes de Wikipedia.
- Anotación de imágenes

De todas estas tareas, las más interesantes para este proyecto, son las basadas en recuperación de objetos CLEF2007, detección de concepto de una fotografía CLEF2008, recuperación de fotografías CLEF2008 y tarea de anotación de CLEF2009.

El objetivo de **Object Retrieval Task CLEF2007**, es obtener una serie de resultados que reflejen cómo de bien funciona la recuperación de objetos desarrollada por el participante, sin información adicional. Se usa para el entrenamiento las imágenes train+val de PASCAL Visual Object Classes Challenge 2006, donde existen 10 categorías, y para el test las imágenes PASCAL VOC Challenge 2006.

La **detección visual por concepto** tiene el objetivo de identificar los conceptos visuales que ayuden en la solución de la tarea de **recuperación de fotografía en ImageCLEF2008**. Las imágenes de la base de datos son etiquetadas de acuerdo con una jerarquía de clases (Figura 2.17), junto con su clasificación y serán estos datos los usados para entrenar el sistema. Posteriormente con una nueva base de datos se tendrá que determinar la presencia o ausencia de estos conceptos.

En la **tarea de anotación y detección visual por concepto** de CLEF2009, se siguió con la idea de ImageCLEF2008 pero con bases de datos de varios miles de imágenes, del conjunto de datos de Flickr MIR 25000 imagen. Todas las imágenes tienen anotaciones múltiples, que se suelen referir a conceptos visuales y holísticos, por ejemplo estaciones, resumen de categorías, personas, etc. En definitiva nos proporcionan las anotaciones de 53 conceptos y se podrán utilizar el orden jerárquico de los conceptos y las relaciones entre ellos para resolver la tarea de anotación [12].

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

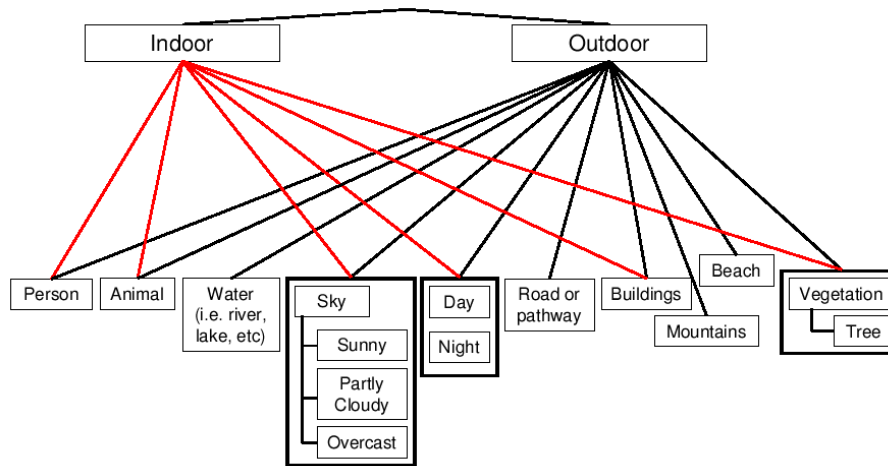


Figura 2.25: Jerarquía de clases de la detección visual CLEF2008

2.3.2 PASCAL VOC

Otro de los foros sobre los que se está trabajando con el reconocimiento y etiquetado de imágenes es PASCAL VOC, Visual Object Classes Homepage. Algunos de sus objetivos son [18]:

- Compilar un conjunto estandarizado de bases de datos de reconocimiento de objetos.
- Proporcionar anotaciones estandarizadas de objetos en todas las bases de datos.
- Proporcionar un conjunto común de herramientas para acceder y gestionar las anotaciones de las bases de datos.
- Ejecutar una evaluación de los resultados de reconocimiento de objetos de clases.

Al igual que pasaba con ImageCLEF, aquí también se van proponiendo unas tareas cada año desde su comienzo en 2005. A los participantes se les proporciona un kit de desarrollo con código y documentación y las bases de datos para el entrenamiento y test. A continuación se hace un breve resumen sobre el contenido de PASCAL VOC para cada año.

VOC2005

Se eligen cuatro clases de objetos:

- Motos, Bicicletas, Coches
- Gente

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Mediante la base de datos de entrenamiento proporcionada, donde cada imagen contiene un archivo de anotación con un cuadro delimitador y una etiqueta de la clase del objeto, se tiene que predecir la presencia/ausencia de esas clases en las imágenes de la base de datos de test, y/o predecir el cuadro y la etiqueta de cada objeto.

VOC2006

Se basa en los mismos conceptos que el año anterior, pero con 10 clases diferentes, y con la diferencia que en la segunda etapa las imágenes de la base de datos no tendrán anotaciones sobre el cuadro ni la etiqueta. Clases:

- Bicicleta, autobús, coche, moto
- Gato, vaca, perro, caballo, oveja
- Personas.

VOC2007

Se aumentan las clases y se agregan unos retos taster para mostrar el interés en la segmentación y el diseño. Las clases se quedaran así:

- Personas: persona
- Animales: pájaro, gato, vaca, perro, caballo, oveja.
- Vehículos: avión, bicicleta, barco, autobús, coche, moto y tren.
- Interior: botella, silla, mesa, maceta, sofá, tv.

VOC2008

Se usan también veinte clases de objetos pero el total de imágenes es menor que en VOC2007. Se añaden también los desafíos taster.

VOC2009

En este año se usan más imágenes en total, se tienen veinte clases y además de las dos tareas que había que implementar, se añaden la competición de la segmentación (separar el fondo de la imagen) y competición del diseño de la persona (predecir el cuadro delimitador y la etiqueta de cada parte del cuerpo de una persona).

3. DISEÑO E IMPLEMENTACIÓN

3.1 ARQUITECTURA DEL SISTEMA

El objetivo de este proyecto es el estudio y la implementación de un sistema automático de detección y etiquetado de fotos.

El empleo de imágenes estáticas tiene el inconveniente de que se están utilizando proyecciones en dos dimensiones (foto) de un objeto que es intrínsecamente tridimensional (cara humana). Esto implica una pérdida de información pero se logra una mayor sencillez en la creación de los algoritmos.

En este proyecto se trata de realizar la implementación de un sistema de detección y etiquetado. Por un lado se usará una serie de imágenes de donde obtendremos una cara detectada y recuadrada. La segunda parte trata con una serie de caras de una base de datos de la cual se obtiene la información necesaria para el entrenamiento del etiquetado. Por último uniremos ambas fases para detectar y describir dicha imagen.

En general, todos los sistemas de detección y etiquetado utilizan la misma secuencia de etapas:

1. Detección de la cara en la imagen.
2. Extracción de características.
3. Etiquetado de la cara (descripción).

A continuación se muestra el diagrama de bloques que contiene dichas etapas, el cual será utilizado en este proyecto, y una breve explicación de cada fase. Para profundizar más en este tema, mirar en el capítulo 2 del estado del arte.

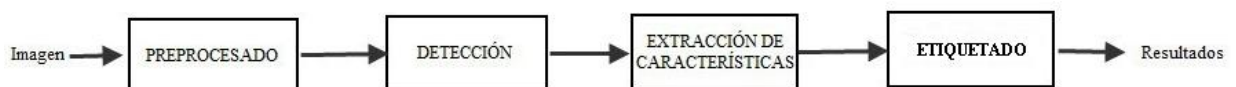


Figura 3.1: Diagrama de bloques del sistema

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- **Preprocesado**

Recibe como entrada una imagen de la base de datos y devuelve dicha imagen tras aplicarle ciertas técnicas, comentadas en el capítulo 2 del estado del arte, que sirven para la mejora de la imagen. En este caso se emplean posibles cambios de tamaño y un filtro para eliminar el ruido.

- **Detección**

Recibe como entrada la imagen preprocesada y devuelve la cara detectada en la imagen, si es que la hay, mediante un recuadro (bounding box). Se presentan dos algoritmos de detección, cada uno basado en una técnica diferente, para posteriormente compararlos.

- **Extracción de características y etiquetado**

Recibe como entrada la cara detectada en un recuadro y sus coordenadas, y devuelve un vector de características. Dichas características son la descripción de la cara en función de ciertos parámetros básicos que nos ayudarán a describir nuestra imagen mediante una serie de etiquetas, como: raza, color de ojos, color de pelo, barba, longitud del pelo y sexo.

3.1.1. HERRAMIENTAS UTILIZADAS

La herramienta de software utilizada para llevar a cabo la implementación de este proyecto es MATLAB 7.0.

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario- GUI).

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Además, se pueden ampliar las capacidades de MATLAB con programas de apoyo especializados, denominados toolboxes. Entre los toolboxes disponibles cabe destacar el de procesamiento de imagen, por el interés que tiene para este proyecto.

3.2. BASES DE DATOS

Como se ha comentado con anterioridad, en este proyecto se van a utilizar diferentes bases de datos de imágenes tanto para comparar los resultados de cada una de las fases del sistema por separado, como para el entrenamiento del algoritmo.

3.2.1. BASE DE DATOS A: FOTOGRAFIAS

En este caso la base de datos está formada por imágenes sacadas manualmente de GoogleImages. Hay un total de 50 fotos, de las cuales 31 son "personajes" españoles famosos y las otras 19 son sacadas aleatoriamente. De todas ellas tenemos:

- 27 imágenes en las que sale una única persona en la foto (13 hombres, 13 mujeres y un niño).
- 13 imágenes de dos personas.
- 5 imágenes en las que salen más de dos personas.
- 5 imágenes de objetos que no tienen cara.

Como podemos observar, en la Figura 3.2 estas imágenes no han sido adquiridas en un ambiente controlado: no se encuentran centradas, cambian los fondos, la luz en cada caso es distinta, etc. A pesar de ello, sí se ha intentado que cumplan, en la mayor medida posible, una serie de condiciones, como son: un tamaño lo suficientemente grande para que se vean las fotos y lo suficientemente pequeño para evitar un elevado tiempo de ejecución, que aparezca parte del cuerpo de las personas, no sólo la cara, y que la persona esté vista de manera frontal (o lo menos girada posiblemente).

Se ha intentado la obtención de imágenes heterogéneas, para comprobar la capacidad de detección de los algoritmos para diferentes casos. Podemos encontrar personas de distintas razas, personas con barba o gorro, que impiden, en cierta medida, la detección de la piel, personas con más parte de piel descubierta que la cara, etc.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes



Figura 3.2: Ejemplo de imágenes de la base de datos A.

3.2.2. BASE DE DATOS B: FOTOGRAFIAS DE CARAS

Esta base de datos está formada por 40 imágenes obtenidas de GoogleImages. Se compone de 35 caras de "personajes" famosos, conocidos en el mundo por cualesquiera motivos, ya sean políticos, actores, músicos, deportistas, empresarios, etc, y por otras 5 imágenes de personas desconocidas.

Como podemos observar, en la Figura 3.3, estas imágenes no han sido adquiridas en un ambiente controlado: no se encuentran centradas, los tamaños de las fotos son diferentes, cambian los fondos, la luz en cada caso es distinta, etc.

A pesar de ello, sí se ha intentado que cumplan, en la mayor medida posible, una serie de condiciones, como son: que aparezca una única cara, que ésta sea una cara entera, que esté vista de manera frontal (o lo menos girada posiblemente) y que ocupe la mayor parte de la foto.

Son un conjunto bastante heterogéneo, ya que se ha pretendido crear un grupo de caras con diferentes características para así poder probar el algoritmo en diferentes casos. Del total de personas hay 18 mujeres y 22 hombres; 33 son de raza blanca y el resto de otras razas; hay diferentes colores de pelo y de ojos; algunos llevan gafas; existen algunos calvos, etc.

A continuación, en la figura 3.3, se muestran algunos ejemplos de estas imágenes.

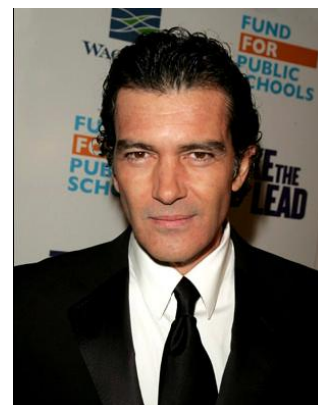




Figura 3.3: Ejemplo de imágenes de la base de datos B

3.2.3. BASE DE DATOS C: TIPOS DE PIEL

Esta base de datos está formada por recortes de piel de imágenes aleatorias obtenidas de Google Images. Se compone de 17 imágenes, de las cuales 5 son de pieles muy claras, 8 de pieles parcialmente morenas y 4 de pieles oscuras. Todos los recortes son extraídos de piel de cara y tienen el mismo tamaño, 16 x 16.

Sirve para entrenar el modelo de detección de caras basado en plantillas y posteriormente para el etiquetado.



Figura 3.4: Ejemplos de imágenes de la base de datos C.

3.2.4. BASE DE DATOS D: TIPOS DE PELO

En este caso la base de datos está formada por 32 imágenes que son recortes de pelo que aparecían en fotografías sacadas aleatoriamente de Google Imágenes. Existen 8 imágenes por cada uno de los cuatro colores básicos de pelo, a saber, moreno, castaño, rubio y pelirrojo. Se ha intentado que dentro de estas 8 imágenes haya diferentes tonalidades de cada color. Además hay que tener en cuenta, que a día de hoy, un gran número de la población se ha cambiado alguna vez el color de pelo, por lo que siempre existen distintos matices que aparecen en la imagen.

Estas imágenes se usaran para el entrenamiento del etiquetado, en las categorías de color de pelo y barba.



Figura 3.5: Ejemplo de imágenes de la base de datos D

3.2.3. BASE DE DATOS E: PLANTILLAS DE OJOS

Existen 4 plantillas de ojos, dos pertenecen al ojo izquierdo y dos al derecho, y cada una tiene diferente color. Se ha pretendido que en estas plantillas, se vea el ojo en su totalidad y que esté lo suficientemente abierto para distinguir el color.

Hay una plantilla del ojo derecho azul y otra marrón, y una plantilla del ojo izquierdo verde y otra negro. Este dato es muy importante debido a que la forma y posición del ojo afectarán al valor que se obtendrá al realizar la correlación de dicha plantilla con el candidato a ojo de la imagen.



Figura 3.6: Imágenes de la base de datos E.

3.3. FASES DEL PROYECTO

Como se ha mencionado anteriormente, este proyecto realiza un estudio y una implementación de un sistema de detección y etiquetado de imágenes. Para ello, se han seguido estos pasos:

- Detección de caras mediante características de bajo nivel con 50 fotos de test de la base de datos A.
- Detección de caras basada en plantillas con 50 fotos de test de la base de datos A.
- Extracción de características y etiquetado, con 40 fotos de la base de datos B y las coordenadas de la situación de la cara en cada una de las imágenes.
- Comparación de resultados entre ambos algoritmos de detección en función de la precisión, la cobertura (recall) y la media ponderada (F-measure).
- Evaluación del algoritmo de etiquetado, para cada una de las etiquetas y para el conjunto el global de ellas.
- Fase final, donde se unen los algoritmos de detección de caras y la extracción de características para el etiquetado. Obtenemos precisión, la cobertura y la media ponderada (F-measure) para la base de datos A.

3.4. PREPROCESADO DE LAS IMÁGENES

Las técnicas de preprocesado pretenden mejorar o realzar las propiedades de la imagen para facilitar las siguientes operaciones, tales como la etapa de detección, la extracción de características y el etiquetado.

Las técnicas de preprocesado se pueden dividir, en general, en función de las pretensiones de sus transformaciones, en algunas de las siguientes facetas [22]:

- Manipulación del contraste (enhancement).
- Eliminación del ruido (denoising).
- Realce de bordes (edges).

A continuación se expondrán las técnicas de preprocesado desarrolladas tanto para el algoritmo de bajo nivel como para el de plantillas.

3.4.1. ELIMINACION DEL RUIDO

En imágenes digitales se considera ruido a cualquier valor de un píxel de una imagen que no se corresponde con la realidad. El ruido se debe, la mayoría de las veces al equipo electrónico utilizado en la captación de las imágenes (ruido de cuantificación de la imagen o efecto de niebla en la imagen) y al ruido añadido en los tramos de transmisión (posible interferencias o errores al transmitir los bits de información [29]).

Existen principalmente dos clases diferentes de ruido:

- Ruido gaussiano: se caracteriza por tener un espectro de energía constante para todas las frecuencias. Este efecto, suma o resta un determinado valor al nivel de intensidad real y es independiente de los valores que toma la imagen. Para su eliminación se tendría que usar un filtro gaussiano, explicado en el capítulo 2 del estado del arte.
- Ruido aleatorio: se caracteriza por la aparición de píxeles con valores arbitrarios que se pueden detectar porque se diferencian mucho de sus vecinos más próximos. Para eliminarlo se tendría que usar un filtro paso bajo o uno de medianas, como se comenta a continuación, en 3.4.1.1 y 3.4.1.2.

3.4.1.1. Filtros de paso bajo

Los filtros de paso bajo enfatizan las frecuencias bajas, que se corresponden con las superficies homogéneas. Tratan de asemejar los niveles de gris de cada píxel a los niveles de gris de los píxeles vecinos, reduciendo la variabilidad espacial de la imagen.

Para el algoritmo de detección de plantilla, que se explica más adelante, usamos un filtro paso bajo donde la máscara de filtrado es la que promedia un entorno de 3x3 píxeles. Se suman los 9 valores y se divide por 9. El peso de cada valor inicial es 1/9 del resultado.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$$=$$

1	1	1
1	1	1
1	1	1

$$\times 1/9$$

Figura 3.7: Máscara de filtrado de 3x3

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Cuanto más grande sea el entorno abarcado, más se difuminan los detalles de la imagen, es por eso que utilizamos el de 3x3, que no desenfocherà demasiado [27].

Como acabamos de comentar, el problema de utilizar filtros paso bajo para eliminación de ruido es que los bordes de los objetos se vuelven borrosos y estos son muy útiles ya que contienen una enorme cantidad de información de la imagen. Para el algoritmo de detección basado en plantillas se decidió usar un filtro paso bajo ya que los bordes aquí no son muy importantes y además este método no conlleva mucho coste computacional.



Figura 3.8: a) Imagen de la base de datos A; b) Imagen segmentada con detector de plantilla; c) Imagen filtrada con el filtro paso bajo.

3.4.1.2. Filtro de medianas

En el filtrado de mediana, el nivel de intensidad de cada píxel se reemplaza por la mediana de los niveles de gris de su vecindad, en lugar de por la media. La mediana M de un conjunto de valores, es tal que la mitad de los valores del conjunto son menores que M y la mitad de los valores mayores que M , es decir en un conjunto ordenado de mayor a menor o viceversa, sería el valor de la posición central [29].

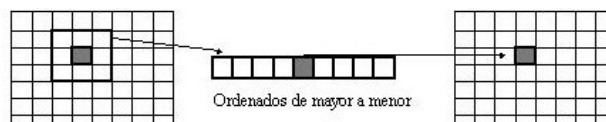


Figura 3.9: Funcionamiento del filtro de medianas.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Numéricamente, un ejemplo podría ser:

$$\begin{pmatrix} 89 & 91 & 20 \\ 87 & 93 & 88 \\ 92 & 86 & 90 \end{pmatrix} \rightarrow \{89 \ 91 \ 20 \ 87 \ 93 \ 88 \ 92 \ 86 \ 90\} \rightarrow \{20 \ 86 \ 87 \ 88 \ 89 \ 90 \ 91 \ 92 \ 93\}$$

Para este caso, la mediana sería el nivel 89. En consecuencia, el ruido que tiene un valor atípico en el entorno (20) quedará colocado en los extremos de la ordenación. La mediana se presenta como una buena técnica para el suavizado en general y en particular para la cancelación de ruido impulsivo. El mayor inconveniente es su alto coste computacional.

Como en la detección de cara de bajo nivel, se usan técnicas más sencillas que en el otro algoritmo, se decidió utilizar el filtro de medianas, ya que en este caso puntual, no existen restricciones de tiempo.



Figura 3.10: a) Imagen de la base de datos A; b) Imagen segmentada con detector de bajo nivel; c) Imagen filtrada con el filtro de mediana.

Implementación

Para la implementación de ambos filtros se utilizan las siguientes funciones de Matlab.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<i>medfilt2</i>	Realiza el filtro de medianas de una matriz de 2D	Matriz de datos	Matriz filtrada
<i>filter2</i>	Realiza un filtro basado en correlación de 2D.	La máscara de filtrado y la matriz de datos	Matriz filtrada

Tabla 3.1: Funciones de Matlab para el filtrado.

3.5. DETECCIÓN DE CARAS

La fase de detección consiste en la segmentación de la imagen en regiones conforme a algún criterio de homogeneidad. En este proyecto se trata de encontrar la región de la imagen en la que aparece una cara humana.

Las principales técnicas de detección de caras humanas en imágenes se han expuesto en el capítulo 2 del Estado del Arte. Aquí se van a presentar, en detalle, las dos técnicas implementadas para el desarrollo del proyecto, que como mencionamos en apartados anteriores, una es la técnica basada en rasgos de bajo nivel y la otra basada en plantillas.

3.5.1. DETECCIÓN DE CARA DE BAJO NIVEL

El primer algoritmo creado pertenece al grupo de técnicas basadas en rasgos de bajo nivel, como pueden ser el color de la piel o la geometría facial. En nuestro caso específico, está basado en el color de la piel.

Este tipo de detección puede tener el problema de que zonas de la imagen que no contengan caras pero que sí tengan un color parecido al de la piel humana, se marcarán como zonas de piel de igual manera. Para intentar determinar qué parte de la imagen es realmente una cara se trata de localizar otras características, como son el factor de forma, la solidez, etc.

3.5.1.1 Detección de píxeles de piel

Se han realizado numerosos estudios sobre la detección de piel a nivel de píxel, algunos de ellos son [10]:

- **"Comparison of five color models in skin pixel classification"**: realizan una comparación de cinco espacios de color y dos métodos no paramétricos de modelado de piel (*lookup table* y modelo de probabilidad de Bayes) [33].
- **"Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images"**: comparan nueve espacios de color y dos métodos paramétricos (modelo de gaussianas y mezcla de gaussianas) [28].

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- "**A comparative assessment of three approaches to pixel-level human skin-detection**": evalúan tres estrategias diferentes de modelado del color de piel [3].

- "**An elliptical boundary model for skin color detection**": comparan dos de los más populares modelos de piel parametrizados en diferentes espacios de color y proponen un modelo propio [14].

A modo de resumen en la siguiente tabla se muestran los resultados de algunos de los métodos utilizados en los artículos anteriores [30].

<u>Método</u>	<u>Referencia</u>	<u>True positive</u>	<u>False positive</u>
Bayes SPM in RGB	[3]	93.4%	19.8%
Elliptical boundary model in CIE-xy	[14]	90.0%	20.9%
Single Gaussian in CbCr	[14]	90.0%	33.3%
Gaussian Mixture in IQ	[14]	90.0%	30.0%
Thresholdin of I axis in YIQ	[3]	94.7%	30.2%

Tabla 3.2: Rendimiento de diferentes detector de piel.

Finalmente para nuestro método se ha optado por la utilización del espacio de color RGB. Varios investigadores consideran que el color de la piel está descrito por las crominancias y no depende de la luminancia, esto haría que el espacio de color RGB quizás no fuera el más adecuado para la detección de piel, ya que mezcla crominancia y la luminancia. A pesar de esto, se opta por su utilización debido a su gran sencillez y los resultados aceptables vistos en la bibliografía consultada, como [3, 19]

La detección de píxeles de piel consiste simplemente en comprobar si cada uno de los píxeles de la imagen cumple unas determinadas condiciones. Si las cumple quiere decir que es un píxel de piel humana. Las condiciones que deben cumplirse para que un píxel sea considerado como de piel humana se muestran a continuación [13]:

$$\begin{aligned} &R > 95 \text{ AND } G > 40 \text{ AND } B > 20 \\ &\text{AND} \\ &\max\{R, G, B\} - \min\{R, G, B\} > 15 \\ &\text{AND} \\ &|R - G| > 15 \\ &\text{AND} \\ &R > G \text{ AND } R > B \end{aligned}$$

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

El resultado de la detección de piel es una imagen binaria (formada por 0s y 1s) en la que cada píxel puesto a uno se considera píxel de piel y el resto son puestos a cero.

La principal ventaja de este método es la simplicidad en las reglas de detección de piel, lo que lleva a la construcción de un clasificador rápido. En el capítulo 4 Resultados de la Evaluación se comparará esta técnica con la basada en plantillas y veremos si compensa la sencillez de este método para reducir el tiempo de ejecución en relación a su efectividad.

A continuación se muestra un ejemplo del resultado de la detección de píxeles de piel en imágenes de la base de datos A.



a)



b)



c)



d)

Figura 3.11: a) c) Imágenes de la base de datos A; b) d) Imagen binaria resultado del detector de piel de bajo nivel.

Implementación

Para este detector de píxeles de piel se utilizan sólo funciones básicas de MATLAB.

3.5.1.2. Filtrado y agrupamiento

Una vez se tiene la imagen binaria se procede a eliminar y agrupar regiones de piel para crear las zonas que van a ser candidatas a ser una cara.

Esto se hace mediante operaciones morfológicas. Este tipo de transformaciones se presentaron en el capítulo 2 del Estado del Arte.

En primer lugar se realiza una erosión de la imagen. La erosión elimina pequeños grupos de píxeles irrelevantes, que tienen poca probabilidad de ser una cara, y que pueden ser considerados como ruido. Es una operación anti extensiva, es decir, la imagen original contiene al resultado de la erosión.

En segundo lugar se realiza una dilatación de la imagen. La dilatación añade puntos del fondo que tocan los bordes de un objeto y rellena agujeros. Es una operación extensiva, es decir, el resultado contiene a la imagen original.

Por último, para no perder los agujeros que se tenían de la imagen binaria (ojos, nariz, etc) y que son muy útiles para obtener información, se multiplica la imagen dilatada por la imagen segmentada.

Es importante resaltar el papel del elemento estructurante elegido. En cuanto a su forma, se ha usado un disco, que tiene la ventaja de que el filtro tenga un comportamiento isotrópico, lo cual es necesario si no conocemos la orientación de la cara.

En cuanto al tamaño del elemento estructurante seleccionado, se ha comprobado experimentalmente. Ante todo, cabe destacar que la heterogeneidad de la base de datos ha complicado la utilización de los operadores morfológicos. En ella aparecen fotos en las que las caras están muy próximas al objetivo, por lo que sería conveniente usar un tamaño del elemento estructurante pequeño, ya que los píxeles de la imagen, tienen muchas probabilidades de que sean caras. Por el contrario, hay otras imágenes donde la mayor parte es paisaje, por lo que convendría un elemento estructurante menor.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

También existen fotos con varias caras muy próximas entre sí, por lo que lo más recomendable es usar un elemento estructurante grande en la erosión para separar lo máximo posible esas caras.

Por este motivo para entrenar al sistema, se probó con diversas fotos sacadas de Internet, para observar la respuesta que se tenía al ir cambiando el elemento estructurante.

Tras realizar varias pruebas con el tamaño del radio, se ha comprobado que tanto para la erosión como para la dilatación los mejores resultados se obtienen con un disco de radio 3.

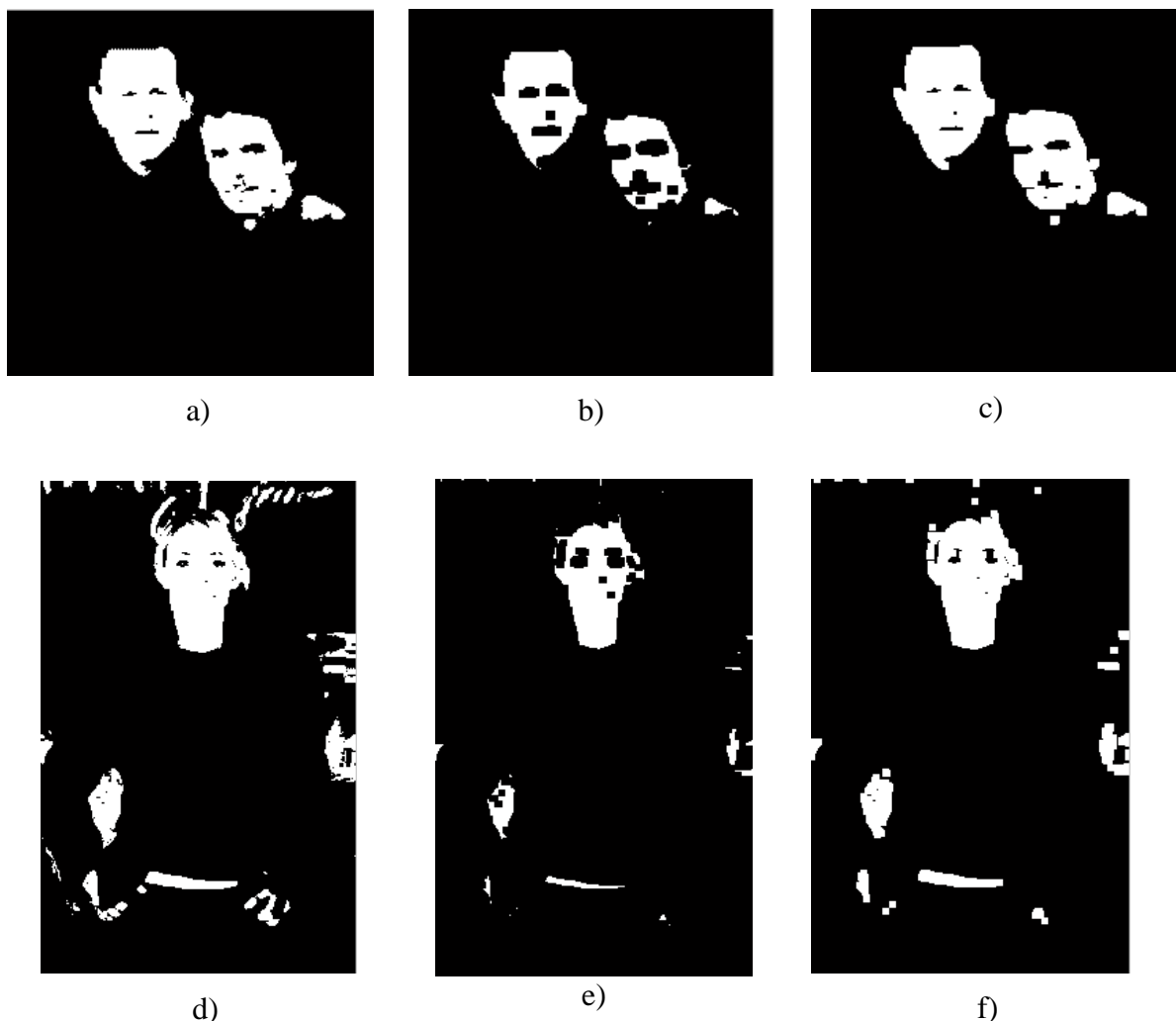


Figura 3.12: a) d) Imágenes segmentadas; b) e) Imágenes tras la erosión de disco 3; c) f) Imágenes tras la dilatación de disco 3.

Después de realizar las operaciones de erosión y dilatación, se procede al etiquetado de las regiones, que pasarán posteriormente a la validación de caras.

Implementación

Se utilizarán las siguientes funciones de MATLAB descritas en la tabla, tanto para el proceso morfológico como para el etiquetado de regiones.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<i>strel</i>	Crea un elemento estructurante, EE	Tipo y radio del elemento estructurante	Elemento estructurante
<i>imerode</i>	Ejecuta una erosión con el EE especificado	Imagen y elemento estructurante	Imagen erosionada
<i>imdilate</i>	Ejecuta una dilatación con el EE especificado	Imagen y elemento estructurante	Imagen dilatada
<i>immultiply</i>	Multiplica dos imágenes	Imagen segmentada e imagen dilatada	Imagen dilatada con agujeros
<i>bwlabel</i>	Etiquetado de objetos en imagen binaria	Matriz a etiquetar	Matriz con las etiquetas

Tabla 3.3: Funciones de MATLAB para el filtrado.

3.5.1.3. Validación de caras

Con las componentes ya etiquetadas se pasa a la parte de validación de caras. Para ello se va a realizar en función de unas ciertas condiciones mínimas que deben tener las regiones de piel para poder ser consideradas como caras.

Existen muchas características que pueden ser tenidas en cuenta a la hora de seleccionar las caras, algunas se pueden ver en el artículo [14]. En este algoritmo las condiciones que se van a utilizar son:

- Factor de forma: mide la regularidad de una región. Las caras son más o menos circulares, por lo que su factor de forma debería tender a 1. Sin embargo al detectar los píxeles de piel es muy probable que también se detecten zonas de cuello por lo que la forma de la región será más parecida a una elipse, por ello el factor de forma se reduce. Probando con diversas imágenes externas a las bases de datos y modificando el valor del factor de forma mínimo que deben tener las regiones para ser consideradas caras, finalmente se ha fijado en 0.1.

$$ff = 4 \cdot \pi \cdot \frac{Area}{Perimetro^2}$$

- Relación de aspecto: es la relación entre el diámetro mayor y el diámetro menor de la región. Haciendo diferentes estudios sobre un conjunto extra de imágenes se obtuvo que el valor mínimo de esta propiedad debe ser 0.3 para las regiones que son caras.

$$ra = \frac{Eje\ menor}{Eje\ mayor}$$

- Número de agujeros: indica el número de agujeros que tiene una región en función de su número de Euler. Ya que buscamos una cara, sabemos que nuestra región tiene que contener varios agujeros, aunque como hemos comprobado experimentalmente, hay veces que estos se eliminan con los operadores morfológicos, por sombras, etc. Por ello se ha elegido que el número mínimo de agujeros sea 1, de tal manera que otras zonas, como ropa, manos o fondo, que no tienen agujeros, sean eliminadas.

$$Agujeros = 1 - N^{\circ} Euler$$

- Solidez: indica qué porcentaje de la región está dentro del cerco convexo de la misma. El cerco convexo es la mínima región convexa que contiene a la región. Se asume que su valor debe ser alto para caras, por lo que su valor mínimo se fija arbitrariamente en 0.7.

Si una región cumple todas estas condiciones, diremos que es una cara.

Implementación

Se utilizan funciones básicas de MATLAB, así como la función para obtener las propiedades de una imagen. Dichas funciones se representan en la Tabla 3.4.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<i>regionprops</i>	Mide un conjunto de propiedades para cada región etiquetada	Matriz etiquetada y propiedades que se quieren	Array con las medidas especificadas

Tabla 3.4: Funciones de MATLAB para validación de caras

3.5.2. DETECCIÓN DE CARA CON PLANTILLAS

El segundo algoritmo creado para este proyecto, pertenece a las técnicas que usan plantillas para la detección de caras. En un principio, este método es más costoso que el anterior, por lo que el tiempo de ejecución debería ser mayor, al igual que sus prestaciones.

3.5.2.1 Detección de píxeles de piel

Como se comentaba en el apartado 3.5.1.1, hay numerosas técnicas para la detección de piel y muchas de ellas consideran que el color está descrito por las crominancias y no por las luminancias, por este motivo, en este algoritmo se usará el espacio YCbCr. El problema del espacio RGB, es que la triple componente representa no sólo el color, sino también la luminancia, la cual puede variar en la cara debido a la iluminación ambiental, por lo que no es una medida fiable. Aunque al usar YCbCr se aumenta la complejidad, se espera ganar en precisión.

Para obtener la piel humana lo primero que se hace es obtener un modelo de piel a través de la base de datos C. Esta está compuesta de 17 recortes de piel de diferentes tonalidades. Con ella se crea una Gaussiana a partir de las crominancias de esos recortes en el espacio YCbCr.

Para cada píxel de nuestra imagen se sacan sus crominancias y se calcula mediante la Gaussiana, la probabilidad de que un píxel sea de piel. Este proceso transforma una imagen en color en una imagen en escala de grises de modo que el valor de gris de cada píxel, indica la probabilidad de que pertenezca a piel.

Tras esto, mediante un proceso de umbral, se obtendría la imagen segmentada. Dado que las personas con pieles diferentes tienen distintas probabilidades de piel, se necesita un umbral adaptativo hasta encontrar el umbral óptimo.

Para ello se crea un primer umbral y si este es menor que el valor de probabilidad de piel se pone a uno. Tras hacerlo con toda la imagen, se suman todos los unos, se aumenta el valor del umbral y se vuelve a realizar la comparación. El umbral óptimo será el punto en el que se obtiene el mínimo de píxeles puestos a uno.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Cuando ya se ha encontrado, usando ese umbral óptimo, segmentamos la imagen, de manera que se obtendría una imagen binaria con los píxeles de piel puestos a 1 y viceversa.

Las principales ventajas de este método, es que usa solo la información de crominancias, desechando las luminancias y además al tratarse como una función Gaussiana, se tendrá una idea bastante acertada de la probabilidad exacta de que sea piel. El inconveniente es que al tener que entrenar el sistema con una serie de plantillas de piel, empeorará la velocidad de respuesta.

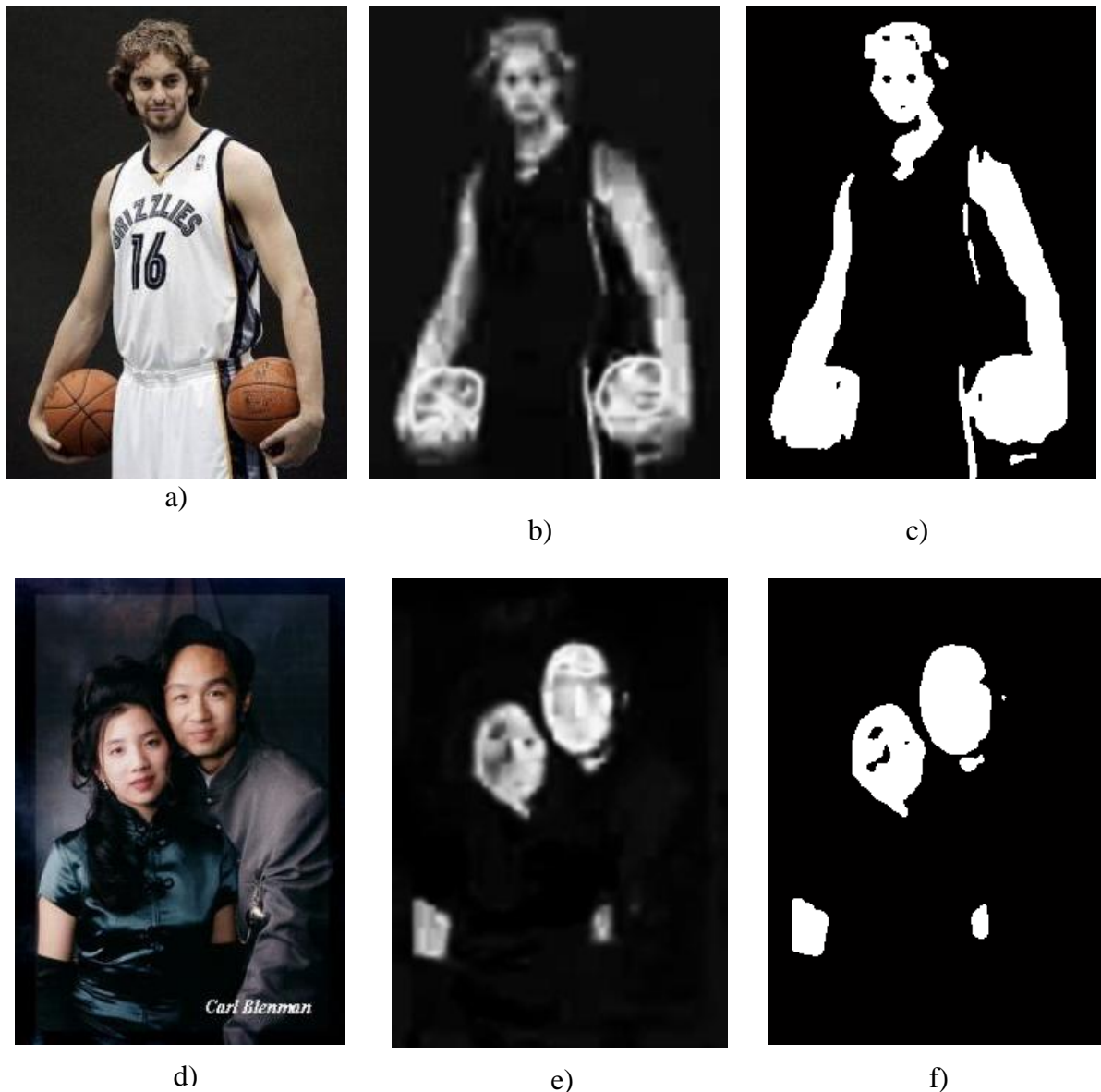


Figura 3.13: a) d) Imágenes de la base de datos A; b) e) Imagen de probabilidad de piel; c) f) Imagen binaria resultado del detector de plantilla

Implementación

Se utilizarán las siguientes funciones de MATLAB, tanto para la creación del modelo de piel, como para la segmentación.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<i>rgb2ycbcr</i>	Convierte una imagen RGB en YCbCr	Imagen en RGB	Imagen en YCbCr
<i>find</i>	Pone a uno los valores que cumplen la condición	Condición	Matriz binaria

Tabla 3.5: Funciones de MATLAB para la segmentación

3.5.2.2. Filtrado y agrupamiento

Al igual que se hizo para el algoritmo de detección basado en rasgos de bajo nivel, ahora pasamos a eliminar y agrupar regiones de piel para crear las zonas que van a ser candidatas a ser una cara. Esto se realiza con operadores morfológicos, ya comentados en el capítulo 2 del Estado del Arte.

En primer lugar se ha realizado una erosión de la imagen. La erosión elimina pequeños grupos de píxeles irrelevantes, que tienen poca probabilidad de ser una cara, y que pueden ser considerados como ruido. Es una operación anti extensiva, es decir, la imagen original contiene al resultado de la erosión.

En segundo lugar se ha realizado una dilatación. La dilatación añade puntos del fondo que tocan los bordes de un objeto y rellena agujeros. Es una operación extensiva, es decir, el resultado contiene a la imagen original.

Es importante resaltar el papel del elemento estructurante elegido. Al igual que pasaba en el otro algoritmo, se ha usado en forma de disco, por los motivos ya explicados en 3.5.1.2.

En cuanto al tamaño del elemento estructurante seleccionado, se ha comprobado experimentalmente. Como se comentó para el otro algoritmo en el apartado 3.5.1.2, al tratarse de la misma base de datos, tenemos mucha heterogeneidad, que complica la tarea de decidir el tamaño correcto del elemento estructurante.

Después de distintos estudios sobre un conjunto de imágenes extra cambiando el tamaño del radio, se ha comprobado que los mejores resultados se obtienen con una erosión de radio 10 y una dilatación de radio 8.

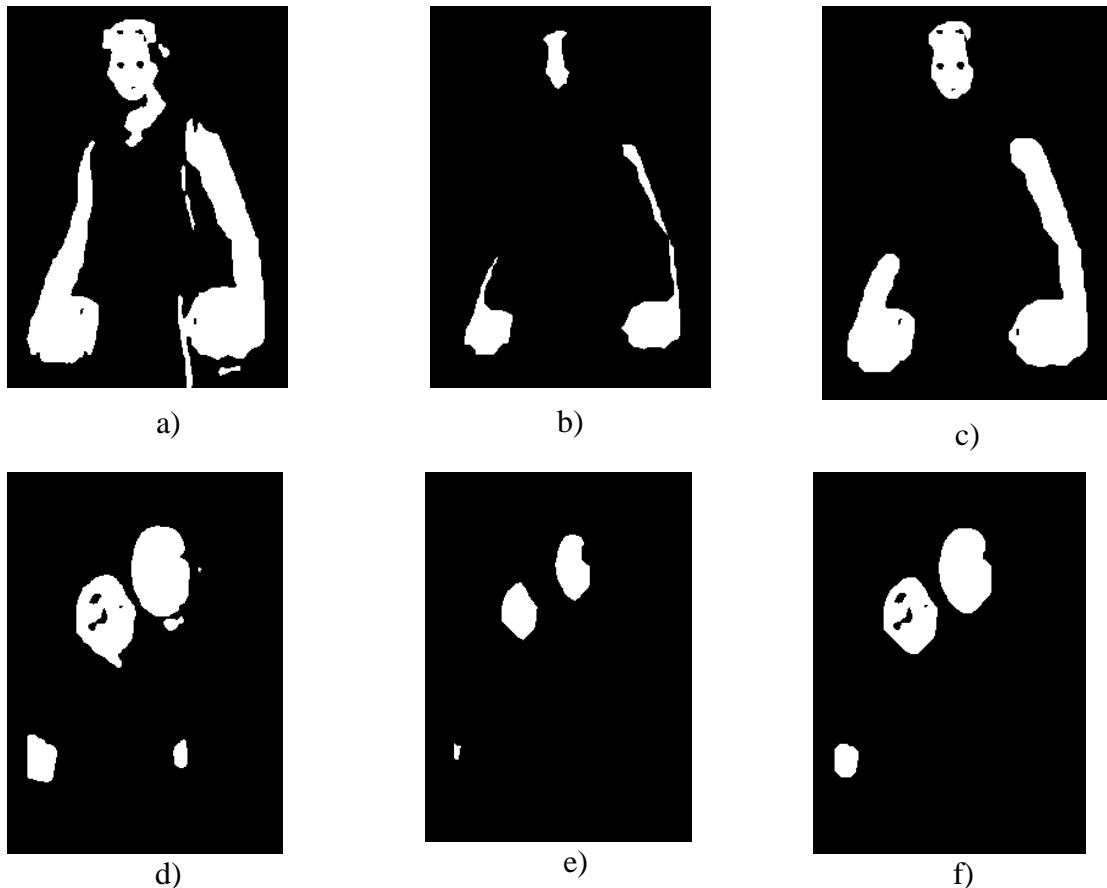


Figura 3.14: a) d) Imágenes segmentadas; b) e) Imágenes tras la erosión de disco 10; c) f) Imágenes tras la dilatación de disco 8;

Una vez se han aplicado estas operaciones morfológicas se procede al etiquetado de las regiones.

Implementación

Para este apartado se usan las mismas funciones que en el algoritmo de detección de caras de bajo nivel, expuesto en la tabla 3.3.

3.5.2.3. Validación de caras

Con las componentes ya etiquetadas se pasa a la parte de validación de caras. Esta se va a realizar en función de unos ciertos test que tendrán que ir pasando las regiones etiquetadas. Estos test se basan en unas condiciones mínimas que deben tener las regiones de piel para poder ser consideradas como caras.

Como se menciona anteriormente, en 3.5.1.3, existen muchas características que pueden ser tenidas en cuenta a la hora de seleccionar las caras. En este método se usa:

- Número de agujeros: indica el número de agujeros que tiene una región en función de su número de Euler. Como ya se comentó en el apartado 3.5.1.3, el número de agujeros para considerar a una región como cara, tiene que ser como mínimo 1.

$$\text{Agujeros} = 1 - \text{N}^\circ \text{ Euler}$$

- Relación de aspecto: es la relación entre el diámetro mayor y el diámetro menor de la región. Tras varios estudios con diferentes imágenes extra, se decidió que la relación de aspecto esté comprendida entre 1 y 3.5.

$$ra = \frac{\text{Eje menor}}{\text{Eje mayor}}$$

Por último las regiones que hayan pasado, tanto el test de euler como el de relación de aspecto, tendrán que cumplir el test de plantilla, para ser consideradas caras.

Ese test implica el uso de una plantilla robotizada de una cara y se trata de modificar esa plantilla en función de la orientación y del tamaño de cada una de las regiones que son candidatas a caras. Cuando ambas imágenes, plantilla y región de piel, están igualmente orientadas y miden lo mismo, se aplica la correlación. La correlación de dos imágenes nos indica cuánto de parecidas son dichas imágenes, cuyo resultado tiende a 1 cuanto más se parezcan.

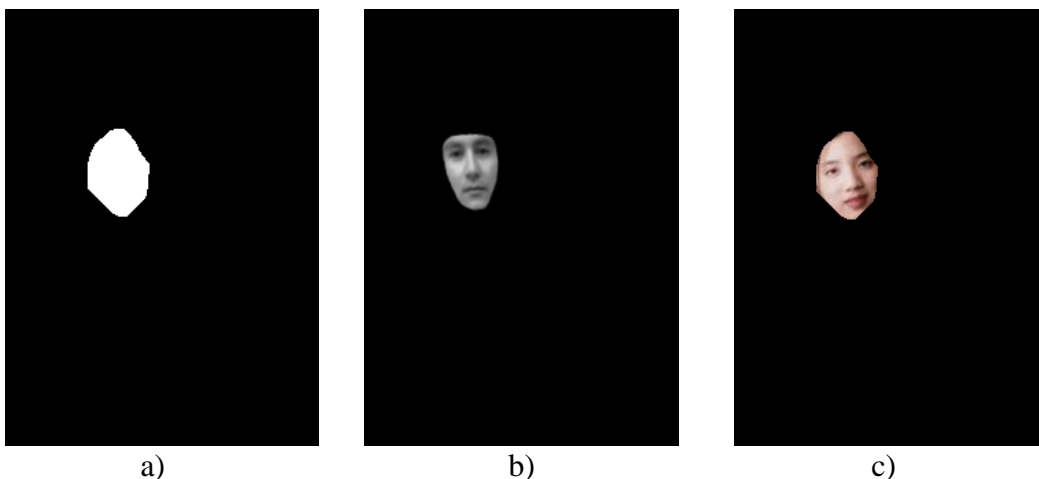


Figura 3.15: a) Región que ha pasado el test de Euler y el de ratio favorablemente; b) Plantilla modificada en función de la región; c) Cara detectada, cuya correlación fue 0.8354.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Experimentalmente se puede comprobar que la región de piel no es exactamente toda la piel de la cara, ya que siempre hay factores externos que afectan (sombras, el pelo que cae por encima de la cara, un fondo de color parecido a la piel, etc.) y además tras su paso por los operadores morfológicos se pueden eliminar píxeles de piel.

Este es el motivo por el que al correlacionar la plantilla y la región de piel, el resultado no es siempre tan exacto como se esperaría. Por ello, se ha decidido arbitrariamente, que una región se considerará cara cuando tenga un valor mínimo de correlación de 0.6.

Implementación

Para este apartado se utilizarán las siguientes funciones de MATLAB, además de algunas de las ya vistas anteriormente.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<i>bwselect</i>	Selecciona objetos de una imagen binaria	Imagen, coordenadas x,y, vecindad	Imagen binaria que contiene al objeto
<i>imfill</i>	Rellena agujeros en una imagen binaria	Imagen	Imagen rellena
<i>imrotate</i>	Rota una imagen	Imagen, ángulo, método, tipo de salida	Imagen rotada
<i>Corr2</i>	Hace la correlación de 2 matrices en 2D	Imagen y plantilla	Valor de la correlación

Tabla 3.6: Funciones de MATLAB para validación de caras

3.6. ETIQUETADO DE CARAS

La fase de etiquetado consiste en la descripción de las caras detectadas en el apartado anterior, conforme a ciertos criterios básicos. Se trata de definir a cada persona de la imagen mediante un vector de características.

Este apartado se ha tratado de forma independiente a la detección de caras, por lo que la entrada que recibe es la imagen y las coordenadas de la cara. Como se comentó en el apartado 3.3, tras el estudio de la fase de etiquetado, se pasará a la unión de ésta con la parte de detección de la cara, de donde se obtendrá directamente las coordenadas empleadas para el etiquetado.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Para el desarrollo de este método se usará la base de datos B, que como se mencionó en el apartado 3.2.2, está compuesta por 40 fotos en las que únicamente existe una cara, centrada y de frente.

Para etiquetar a una persona, se necesita un conjunto de características que describan dicha cara. La biometría es ese conjunto de medidas que se pueden tomar de los rasgos de las personas.

Existen características intrínsecas, que son aquellas con las que el individuo nace y no pueden ser modificadas fácilmente, como por ejemplo, la raza, la forma de la cara, la distancia entre los ojos, el patrón de voz, etc. Y luego existen las características extrínsecas, que son aquellas que pueden variar con el tiempo, como el color de pelo, el uso de piercing, el grosor de las cejas, etc. Ya se han creado múltiples estudios sobre medidas tales como el iris, la huella dactilar, etc, las cuales describen de manera única a cada persona, por lo que en el presente proyecto, no se quiere ir tan lejos, sino que se pretende obtener características que puedan ser apreciables por todo el mundo, aun sin conocimientos de biometría, en un solo vistazo.

Por este motivo se ha elegido una serie de medidas tanto intrínsecas como extrínsecas, que se consideran necesarias y mínimas para describir una cara correctamente en el contexto de este proyecto. Éstas son:

- Raza
- Color de ojos
- Color de pelo
- Barba
- Longitud del pelo.
- Sexo

Como se comentará en el capítulo 5 de conclusiones y líneas futuras, se podrían añadir otros rasgos descriptores de una persona, como el uso de pendientes, sombreros, altura, peso, etc. Para esta primera versión se ha optado por estas seis características, que son lo suficientemente importantes como para crearse una visión acertada de la persona que aparece en la fotografía.

3.6.1. RAZA

La primera característica que se va a tratar es la raza de una persona, una propiedad intrínseca y no modificable. En un principio se pensó obtener la raza caucásica, negra y asiática. Esto planteó un grave problema ya que a pesar de que a la vista existen múltiples diferencias entre todas ellas, en el tratamiento de imágenes tenemos una información demasiado reducida como para obtener resultados aceptables.

Además hay que tener en cuenta que hoy en día hay una componente multirracial muy elevada, por ejemplo la aparición de mulatos (entre raza blanca y negra), la unión entre chinos e indios (chindian) en Malasia, la mezcla producida en Hawai tanto de europeos, como africanos e indios, etc [17].

Al pensar en las diferencias físicas de todas ellas, se llegó a la conclusión de que su mayor rasgo era el color de la piel. El primer intento de obtener la raza, fue por ello, buscar un umbral que diferenciara unas de otras. Se creó un algoritmo para el espacio de color RGB, donde se comprobó que:

- Para todas las razas la componente roja es siempre mayor, en media, que las otras dos componentes.
- En la raza negra, las componentes verde y azul, en media, son menores que para la raza blanca.
- En la raza asiática, el color de la piel no es un rasgo característico, ya que hay muchas personas de esta raza con pieles muy claras y otras con pieles más oscuras. Podría decirse que es un mito el hecho de que los asiáticos tengan la piel amarilla.

Este último motivo, ocasionado por la mezcla de razas comentada antes, fue el que hizo tener que buscar otra idea para la detección de la raza asiática. Entonces se probó con la característica de los ojos rasgados. Para ello se empleó la relación de aspecto, explicado en el 3.5.1.3, donde se debería obtener un valor pequeño promovido por la diferencia entre el eje mayor y el menor del ojo.

Tras varias pruebas tanto para personas asiáticas como caucásicas y negras, se llegó a la conclusión de que los valores obtenidos eran arbitrarios.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Finalmente se optó por etiquetar solo los colores de la piel, es decir, personas con piel blanca o con piel negra. Como se vio con el espacio de color RGB, al usar este las tres componentes de color, sin separar la luminancia, dificulta la tarea de diferenciar entre raza negra y blanca. Se decidió tomar otro espacio de color en el que la luminancia este separada de las crominancias, ya que esta primera es que la nos dice la claridad de la imagen. Por tanto, si tenemos una imagen negra, su luminancia debería tener unos valores muy bajos, próximos al cero, mientras que en una imagen blanca, su luminancia debería ser mayor, rondando el valor 255.

Para comprobar esto, se calculó el histograma de la luminancia empleando la base de datos C, donde se tienen recortes de pieles de diferentes razas y tonalidades.

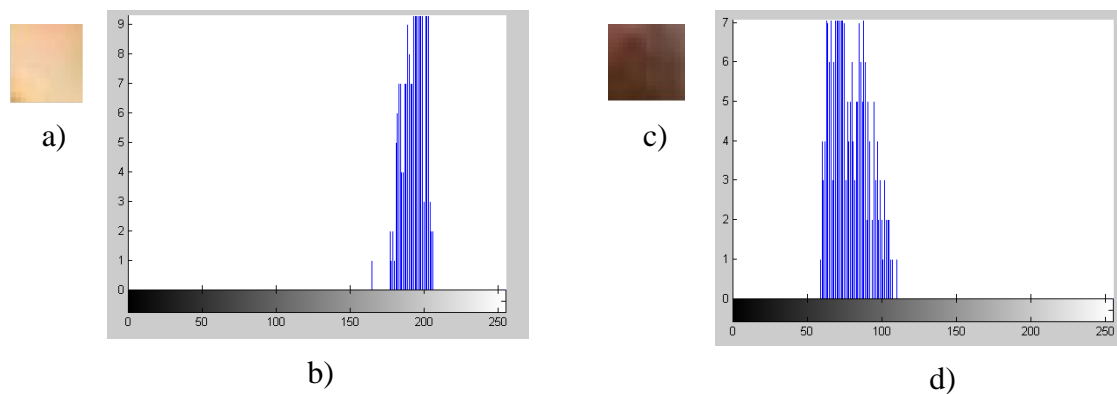


Figura 3.16: a) c) Recortes ampliados de piel; b) d) Histograma de Y.

La Figura 3.16 a) es una piel clara y su histograma de luminancia así lo refleja, con un valor medio de $Y=193.94$. En el caso de la Figura 3.16 c) se tiene una piel negra con un valor medio de $Y=78.0078$.

Como se puede observar, ni la piel blanca, es absolutamente blanca (tiende a un tono rosáceo) ni la piel negra es absolutamente negra (tiende a un tono marrón). Este es el motivo por el que los valores medios de luminancia no dan ni tan altos ni tan bajos como se esperaba.

Para nuestras imágenes de la base de datos B, se ha optado por realizar el histograma de la luminancia en una única zona de la piel, para así crear un histograma más uniforme con solo piel, al igual que los que se usan con la base de datos C. Se ha elegido un recorte de la zona de la frente, ya que es una de las regiones más grandes de piel de la cara. Este recorte no es de tamaño fijo, sino variable, ya que cada imagen tiene diferentes dimensiones.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

De esta zona de la cara, que se considera que será piel, se calcula el valor medio de la luminancia para posteriormente compararlo con los histogramas obtenidos por los recortes de piel de la base de datos C. Estos histogramas se hacen solo con partes que son realmente piel, mientras que en nuestros recortes de las caras de la base de datos B, puede introducirse algo del pelo del flequillo, o un trozo de ceja, etc. Por este motivo, se ha entendido que los histogramas de las regiones de piel de las caras de la base de datos B, no serán tan “puros”.

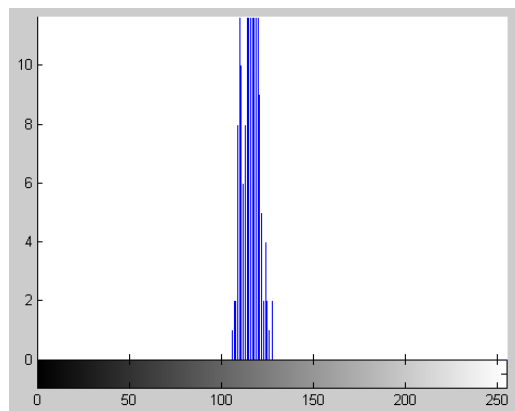
<u>Descripción</u>	<u>Umbrales</u>	<u>Etiquetas</u>
Coger el recorte de la frente, calcular su histograma y obtener la media de Y.	$\text{mediaY} \geq 145$	Blanco
	$\text{mediaY} < 145$	Negro

Tabla 3.7: Posibles valores del etiquetado de raza.

En función de la condición que cumpla nuestro recorte de la cara, se etiquetará de una manera u otra, como ve en la Figura 3.17.



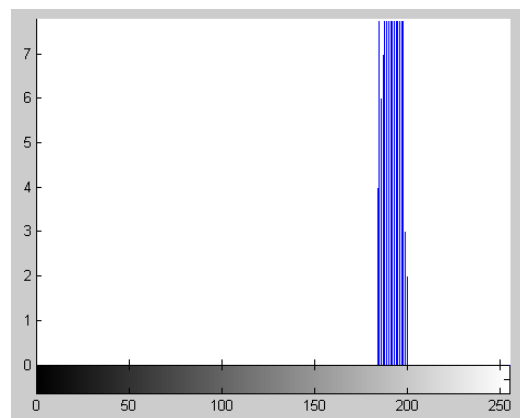
a)



b)



c)



d)

Figura 3.17: a) c) Cara con recorte de piel; b) d) Histograma de Y

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

La Figura 3.17 a) es una cara de raza negra, con un valor medio de $Y=116.3008$, de manera que según los umbrales establecidos se etiquetará como "Piel Negra". En el caso de la Figura 3.17 b) se tiene una cara de raza blanca, con un valor medio de $Y=192.3242$, es decir, se etiquetará como "Piel Blanca".

Implementación

En la Tabla 3.8 aparecen las funciones usadas de MATLAB para la parte de etiquetado de raza.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<i>imcrop</i>	Recorta una imagen	Imagen, coordenadas xy, ancho y alto del recorte	Imagen recortada
<i>imhist</i>	Pinta un histograma	Imagen en escala de grises	Histograma de la imagen

Tabla 3.8: Funciones de MATLAB para etiquetado de raza

3.6.2. COLOR DE OJOS

Otra característica que se consideró necesaria para la descripción de una cara, es el color de ojos.

Como se menciona anteriormente, para la fase de etiquetado, se le pasa una imagen de la base de datos B y las cuatro coordenadas pertenecientes a la cara. Una vez tenemos la cara enmarcada, se necesita localizar los ojos dentro de ella para luego sacar su color.

Para ello se realiza una segmentación basada en el espacio de color RGB, donde los píxeles considerados como piel serán puestos a uno, y el resto a cero. Esta técnica está expuesta en el apartado 3.5.1.1. y se utiliza en contrapartida a la teoría de plantillas, por su sencillez y su rapidez.

Con esto se crea una imagen binaria donde se pueden discriminar perfectamente las zonas de piel de las que no son piel. Como nuestro interés ahora mismo es una zona que no es piel, se crea la imagen inversa, para así diferenciar mejor los ojos a la hora de etiquetar las distintas regiones.

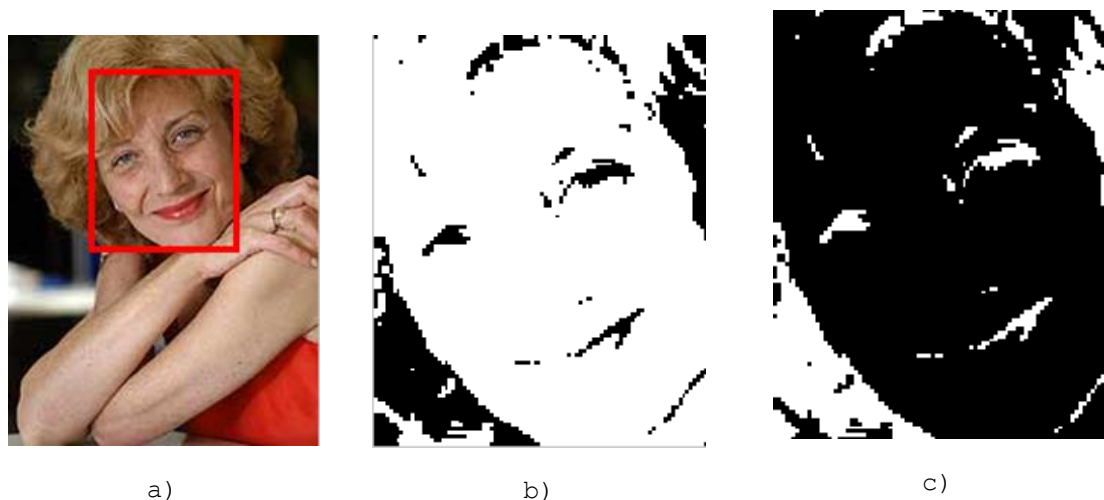


Figura 3.18: a) Imagen con cara recuadrada a mano; b) Segmentación en función del espacio de color RGB; c) Imagen inversa

Una vez se tienen las regiones etiquetadas, se pasa a comprobar si estas podrían considerarse ojos. Esto se realiza mediante ciertas condiciones basadas en las propiedades explicadas en el capítulo 3.5.1.3. Para el caso que nos concierne ahora, nos interesan las características del factor de forma y del área. Recordar que el factor de forma nos indica la regularidad de una región mientras que el área nos indica su tamaño.

Al tratarse de buscar un ojo, y tras varias pruebas, se consideró apropiado no tomar un umbral fijo, sino uno variante en función del tamaño de la cara. Por este motivo se eligió que el área debería ser mayor que un tercio del ancho de la cara.

La otra condición que tiene que cumplir para poder considerarse un ojo, es que el factor de forma sea mayor que 0.2. Se llegó a este valor, tras realizar múltiples pruebas, sabiendo que cuando más cercano este de 1, mayor forma circular tendrá y según se va reduciendo dicho valor se va pareciendo más a una elipse.

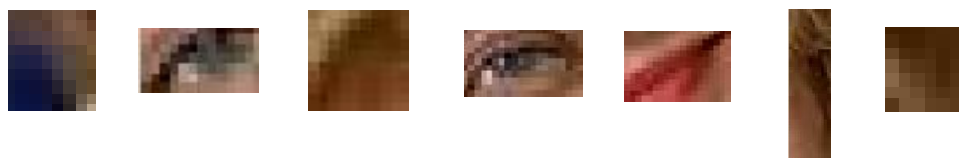


Figura 3.19: Candidatos a ojo correspondientes a la figura 3.17

Cuando ya se tienen los posibles candidatos a ojos se van recortando uno a uno y se descompone en sus 3 componentes de RGB. Una vez conseguido esto, se pasa a correlacionarlo con las plantillas de ojos que forman la base de datos E.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Estas plantillas se redimensionan en base a la medida de la región candidata a ojo, también en su forma RGB. Hay que tener en cuenta que dichas plantillas corresponden a los cuatro colores básicos de ojos y que dos de ellas corresponden al ojo izquierdo y dos al derecho. En el caso de la plantilla de color marrón y de color azul, muestra un ojo derecho y en el caso de las plantillas de verde y negro, muestra el izquierdo. Este es un motivo muy relevante a la hora de hacer la correlación, la cual mide el parecido entre dos matrices, de tal manera, que siempre será mayor cuando el ojo de cada lado coincida con la plantilla de ese mismo lado.

Al hacer las tres correlaciones, una para componente de la imagen, R, G y B, se considera que un ojo es de un determinado color, cuando la suma de las tres correlaciones es la mayor y siempre que dicha suma supere el valor de 0.90. Este umbral ha sido elegido tras la realización de varias pruebas.

<u>Descripción</u>	<u>Umbrales</u>	<u>Etiquetas</u>
Correlación entre la región y cada plantilla.	Correlación con plantilla negra es la mayor	Negro
	Correlación con plantilla marrón es la mayor	Marrón
	Correlación con plantilla verde es la mayor	Verde
	Correlación con plantilla azul es la mayor	Azul

Tabla 3.9: Posibles valores del etiquetado de color de ojos.

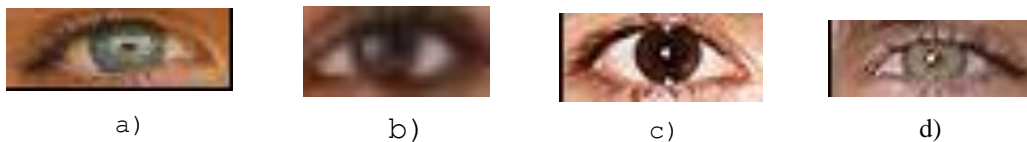


Figura 3.20: Plantillas de: a) ojo derecho azul; b) ojo izquierdo negro; c) ojo derecho marrón; d) ojo izquierdo verde.

Tras realizar la correlación de todos los candidatos a ojos de la Figura 3.19 con las cuatro plantillas de la Figura 3.20, para nuestro ejemplo, se obtuvo que el máximo valor de las tres correlaciones sumadas fue de 1.4455 con la plantilla de ojo verde. Como este resultado es superior al umbral de 0.90 establecido, dicha imagen se etiquetará como "Ojos Verdes".

Implementación

Para el etiquetado de color de ojos, se utilizan funciones ya explicadas en tablas anteriores, como por ejemplo regionprops, imcrop, corr2, etc.

3.6.3. COLOR DE PELO

La siguiente característica que se emplea en el etiquetado de caras, es identificar el pelo y definir su color o la ausencia de él. Las etiquetas que se utilizan son: moreno, rubio, castaño, pelirrojo y calvo.

Una vez que tenemos la cara detectada manualmente, es muy fácil saber donde se encuentra el pelo, por este motivo elegimos el rectángulo superior de la cara para analizarlo posteriormente. Como cada imagen tiene una cara de unas ciertas dimensiones, la zona que se recorta no es fija, sino que depende del tamaño de la cara en cuestión. Se utiliza el mismo ancho, pero solo un cuarto de la altura, situándose esta, por encima de los límites de la cara. Aunque a esta zona se la considera pelo, no tiene porque ser así en su totalidad, ya que podría darse el caso de que tenga poca altura o de que se tratase de una persona calva.

Hay que tener en cuenta, que aunque el desarrollo se esté trabajando sobre unas imágenes con la cara detectada manualmente, es decir, que están perfectamente recuadradas, más adelante nos encontraremos también con ciertos problemas, como por ejemplo, que no se haya cogido como cara la cara entera, sino dejando parte de la frente fuera de la zona seleccionada, o que personas con el pelo muy parecido a la piel (personas rubias) se extienda la zona de la cara incluyendo el pelo.

Cuando ya se tiene la zona que se considera pelo, se tiene que comprobar el color predominante en dicha zona, la media, y compararlo con unas plantillas de pelos de diferentes colores. Para este caso se ha utilizado el espacio de color HSV, explicado en el capítulo 2.1.2.3. del Estado del Arte, que trata por separado la luminancia de las crominancias.



a)



b)

Figura 3.21: a) Imagen con cara detectada; b) Zona de pelo recortado

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En la base de datos D, se tienen 32 plantillas, ocho por cada uno de los cuatro colores básicos que se han utilizado en dicha base. La manera de tratar estas plantillas, es buscando el máximo y el mínimo de cada una, y calculando la media de todas ellas. Así se obtienen unos umbrales con los que se crean las condiciones para ser moreno, rubio, castaño o pelirrojo. Cuando los valores de la zona de pelo cumplen alguna de estas condiciones, se puede decir que pertenece a esa categoría.

Existe también otra condición que se crea a partir de la base de datos C, que son 17 plantillas de piel, para aquellas personas que son calvas. En este caso, se saca la media para cada una de las componentes de HSV de las plantillas de piel y se elige el valor mínimo y máximo de cada una. Cuando los valores de la zona de pelo cumplan esta condición, se dirá que la persona es calva.

<u>Descripción</u>	<u>Umbrales</u>	<u>Etiquetas</u>
Se coge la zona de pelo, que es la zona superior de la cabeza, se obtiene la media de H, S y V y se compara con las condiciones obtenidas de las plantillas de la base de datos D.	Si condición moreno	Moreno
	Si condición rubio	Rubio
	Si condición castaño	Castaño
	SI condición pelirrojo	Pelirrojo
	Si condición piel	Calvo

Tabla 3.10: Posibles valores para el etiquetado de color de pelo.

Para el ejemplo situado arriba, de la figura 3.21, se recorta la zona de pelo y sus valores medios de H, S y V, son tales, que cumplen la condición de pelo castaño, por lo que su etiqueta será "Pelo Castaño".

Implementación

Aparte de las funciones de MATLAB básicas y las ya mencionadas en apartados anteriores, como por ejemplo, `imcrop`, se emplearan también las descritas en la Tabla 3.11.

<u>Nombre</u>	<u>Descripción</u>	<u>Entrada</u>	<u>Salida</u>
<code>rgb2hsv</code>	Convierte una imagen RGB en HSV	Imagen en RGB	Imagen en HSV

Tabla 3.11: Funciones de MATLAB para etiquetado de color de pelo

Para el etiquetado de barba y de la longitud del pelo que se detallarán en apartados posteriores, 3.6.4 y 3.6.5, se aplicarán las mismas funciones que para el etiquetado de pelo, debido a que su implementación es muy similar.

3.6.4. BARBA

Después de hablar del color del pelo, se tiene que hablar de la barba, un dato muy característico, aunque variante, y que a la vez será de gran ayuda a la hora de detectar el sexo.

Al igual que se hizo con el color del pelo, la barba se tratará de la misma manera. Teniendo la cara detectada a mano, se puede localizar de una manera sencilla la barbilla. Esta se situará de manera variable en función del tamaño de la cara de cada imagen y se colocará en el medio de la parte inferior de la cara. Se convertirá la zona de la barbilla al espacio de color HSV y se obtendrán los valores medios para las tres componentes de color.

Como se realizó para el caso del color de pelo, aquí también se necesita de las plantillas de pelo y de piel para obtener los umbrales para la clasificación. Cuando nuestra imagen cumpla alguna de las condiciones de color, sabremos que tiene barba, mientras que si por el contrario cumple la de piel, sabremos que no tiene.

<u>Descripción</u>	<u>Umbrales</u>	<u>Etiquetas</u>
Se coge la zona la barbilla, se obtiene la media de H, S y V y se compara con las condiciones de color de pelo.	Si condición piel	Sin barba
	Si condición moreno o rubio o castaño o pelirrojo	Con barba

Tabla 3.12: Posibles valores del etiquetado de barba.



a)

Figura 3.22: a) Imagen con la cara detectada a mano; b) Zona de la barbilla recortada.



b)

En la figura 3.22 puede verse como se recorta la zona de la barbilla y sus valores medios de H, S y V, son tales, que cumplen alguna de las condiciones de pelo, por lo que su etiqueta será "Tiene Barba".

Implementación

Las funciones de MATLAB empleadas para el etiquetado de la barba son, como se menciona en el apartado 3.6.4, las mismas que las de la tabla 3.11.

3.6.5. LONGITUD DEL PELO

La longitud del pelo tampoco es una característica invariante con el tiempo, pero sí es un dato bastante relevador sobre una persona y además será de gran utilidad a la hora de detectar el sexo.

De la misma manera que se obtiene la zona de pelo y la de la barbilla, se obtiene la zona de donde debería estar el pelo si fuera largo. Se han creado dos regiones iguales, una a cada lado del cuello, ya que como se demostrará más adelante, por motivos de inclinación de la cabeza o por el tipo de peinado de la persona, no siempre existe pelo a los dos lados. Igual que en los casos anteriores se recorta de manera variable en función del tamaño de la cara.

Una vez se tienen ambas zonas se procede a obtener los valores medios de HSV y a compararlos con las condiciones de color. Con que alguna de ambas regiones cumpla alguna de las condiciones de pelo, se etiquetará como que tiene el pelo largo.

<u>Descripción</u>	<u>Umbrales</u>	<u>Etiquetas</u>
Se cogen zonas laterales del cuello, se obtiene la media de H, S y V y se compara con las condiciones de color de pelo.	Si condición moreno o rubio o castaño o pelirrojo	Pelo largo
	Si no cumple ninguna condición	Pelo corto

Tabla 3.13: Posibles valores del etiquetado de la longitud de pelo.

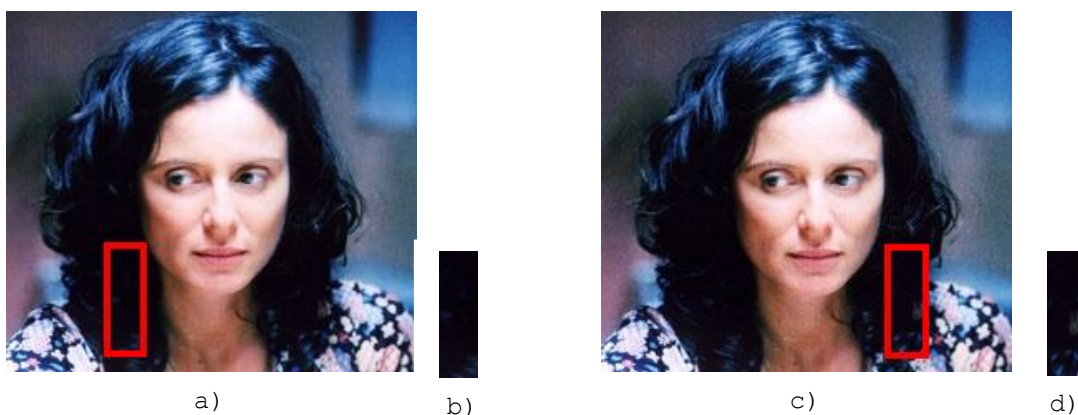


Figura 3.23: a) c) Imagen con la región de pelo recuadrada;
b) d) Imagen recortada de la zona de pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Como se puede observar en la Figura 3.23, se recortan ambas zonas de pelo y sus valores medios de H, S y V, son tales, que cumplen alguna de las condiciones de pelo, por lo que su etiqueta será "Pelo Largo".

Implementación

Las funciones de MATLAB empleadas para el etiquetado de la barba son, como se mencionó en el apartado 3.6.4, las mismas que las de la tabla 3.11.

3.6.6. SEXO

Por último se encuentra otra de las características intrínsecas de una persona, que nos permitirá clasificarla con mayor facilidad, el sexo.

Aunque aparentemente las diferencias físicas entre hombres y mujeres sean muy claras, cuando se trata de visualizarlo a través de matrices, el asunto se complica.

En un primer momento se pensó que un rasgo característico era la nuez, pero tras varias pruebas erróneas se llegó a la conclusión de que no hay ninguna manera de detectar el abultamiento de la nuez en una imagen de dos dimensiones. El mismo resultado se obtuvo al tratar de discriminar el sexo basándose en el grosor de las cejas, que a pesar de que sí existe una diferencia significativa en el tamaño y la forma, según sean hombres o mujeres, no se pudo crear ningún patrón al que se asemejara dicha información.

Por este motivo, se decidió usar un conjunto de medidas, que por separado no aseguraban la diferenciación entre sexos, pero al unir las todas se puede afirmar con bastante probabilidad la pertenencia a una u otra clase.

Las propiedades usadas son la ausencia o presencia de barba, ser calvo, la longitud del pelo y un factor de sexo que se explica más abajo.

- **Barba:** es un rasgo masculino bastante relevante pero no es inequívoco, pues la ausencia de la barba no conlleva que se trate de una mujer.
- **Calvicie:** también es una característica bastante importante, ya que sabemos que si es calvo, será un hombre, pero de igual manera que antes, no podemos afirmar lo contrario.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- **Pelo largo:** en este caso sólo podemos hablar de la probabilidad de que sea de un sexo u otro. Esto es así, ya que existen múltiples hombres con pelo largo y otras tantas mujeres con pelo corto. Además también se puede dar el caso de que lo llevan recogido en la foto, de manera que se detecta como si fuera corto.
- **Factor de sexo:** esta característica se ha creado a partir del factor de forma y de la relación de aspecto. Como se ha explicado en 3.5.1.3, el factor de forma mide la regularidad de la región y la relación de aspecto mide la proporción entre el eje mayor y menor de región. Experimentalmente se ha comprobado que en las mujeres estas dos medidas son algo menores que en los hombres. Por regla general una mujer tendrá la cara más estrecha que un hombre, es decir la relación entre sus ejes será mayor, por lo que la relación de aspecto será más pequeña.
Para el caso del factor de forma, cuando más tienda éste a 1, más se parecerá a un círculo. Como dijimos antes, al tener los hombres la cara más ancha, se parecerá más a un círculo, por lo que el factor de forma tendrá también que ser mayor. Por ello se estipula que una cara pertenece a una mujer cuando el factor de forma es menor que 0.2 y cuando la relación de aspecto es menor que 0.7, siempre que ambos factores sean diferentes de 0.

<u>Descripción</u>	<u>Umbrales</u>	<u>Etiquetas</u>
Se utilizan las etiquetas anteriormente explicadas	Si cumplen las condiciones en cada caso.	Calvo Con/ sin Barba Largo/corto
Se obtiene el factor de sexo, que incluye el factor de forma y la relación de aspecto.	Factor forma > 0.2 Relación de aspecto < 0.7	Factor de sexo=1; Mujer
Se obtiene el factor de sexo, que incluye el factor de forma y la relación de aspecto.	Factor forma < 0.2 Relación de aspecto > 0.7	Factor de sexo=0; Hombre

Tabla 3.14: Posibles valores del etiquetado de sexo.

Una vez ha quedado claro qué características se usan para etiquetar el sexo, solo queda explicar cómo se aplican.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Para este caso no necesitamos las coordenadas de la cara detectada, sino que usamos el método de detectar caras de bajo nivel explicado en 3.5.1, que se basa en el espacio de color RGB. Se aplica una segmentación y los correspondientes operadores morfológicos, y se considera cara a todo aquella región que cumpla las condiciones establecidas. Estas, como se mencionó en el otro apartado, están basadas en el factor de forma, la solidez, la relación de aspecto y el número de Euler. Cuando se detecta una de esas regiones como candidata a cara, se guardan los valores de factor de forma y relación de aspecto, y si cumplen las condiciones para ser mujer, se pondrá el factor de sexo a uno, y viceversa.

De esta manera con las etiquetas explicadas en los apartados anteriores, se tienen ya todas las medidas, que unidas, nos permitirán etiquetar el sexo de la cara. La forma de tratar todas estas características es mediante la aplicación de la lógica. Si se tiene una cara y se obtiene que es calvo, con barba, el factor de sexo sale masculino y tiene el pelo corto, entonces se dirá que es un hombre.

Hay que tener en cuenta, que puede haber etiquetas equivocadas que afectaran al resultado del sexo, por ejemplo, si me dice que una cara de una mujer tiene barba, este valor repercutirá en la toma de decisión para etiquetar el sexo. Se ha pretendido, que cuando haya un error de este tipo, se vea compensado por los otros factores y no se tenga muy en cuenta a la hora de etiquetar, de tal manera que aparecerá que la imagen es probablemente un hombre o una mujer. Cuando se tienen ya dos errores de las cuatro medidas tomadas, por ejemplo en la barba y el factor de sexo, se guardará como un error de etiquetado.

La tabla de la verdad que se aplica para detectar el sexo en función de las cuatro medidas explicadas se muestra en la Tabla 3.15.

<u>Calvo</u>	<u>Barba</u>	<u>Fact. Sexo</u>	<u>Pelo Largo</u>	<u>Resultado</u>
0	0	0	0	Hombre
0	0	0	1	Probablemente mujer
0	0	1	0	Probablemente mujer
0	0	1	1	Mujer
0	1	0	0	Hombre
0	1	0	1	Probablemente hombre
0	1	1	0	Probablemente hombre
0	1	1	1	Error del etiquetado

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

1	0	0	0	Hombre
1	0	0	1	Error de etiquetado
1	0	1	0	Probablemente hombre
1	0	1	1	Error de etiquetado
1	1	0	0	Hombre
1	1	0	1	Hombre
1	1	1	0	Error de etiquetado
1	1	1	1	Error de etiquetado

Tabla 3.15: Tabla de la verdad para el etiquetado de sexo

Las etiquetas de probablemente hombre y probablemente mujer que aparecen en la Tabla 3.15, que como ya se mencionó valen para intentar minimizar los errores de las otras etiquetas, a la hora de la evaluación serán tomadas como si fueran Hombre y Mujer respectivamente.

En la Figura 3.24 se muestra un ejemplo de etiquetado de sexo para una imagen de la base de datos B, así como los resultados obtenidos.

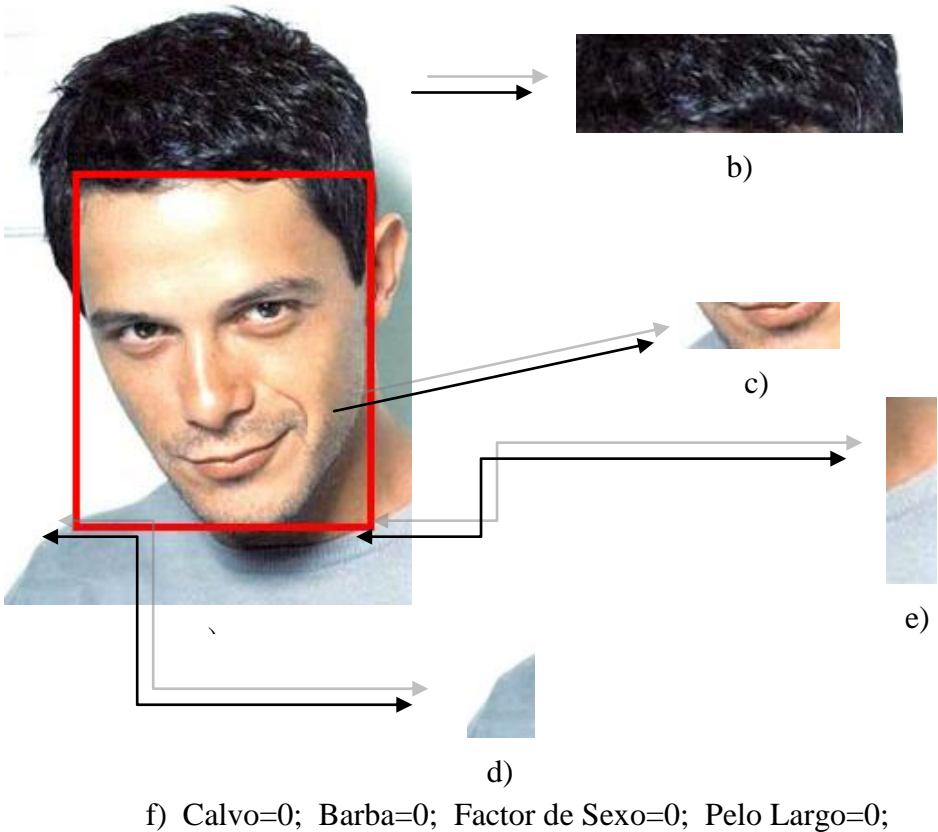


Figura 3.24: a) Imagen con la cara recuadrada; b) Pelo recortado; c) Zona de la barbilla recortada; d) e) Zonas de pelo largo recortadas; f) Resultados del etiquetado de sexo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Con los valores que aparecen en la Figura 3.24 f), si se mira en la Tabla 3.15, el resultado 0 0 0 0, se ve que se le pondría la etiqueta de "Hombre".

Implementación

Para la parte de etiquetado de sexo, como se comentó anteriormente se unen varias etapas de procesos anteriores, por lo que las funciones de MATLAB que se emplean son: para la parte de preprocesado, se usa la Tabla 3.1; para la parte de filtrado morfológico y segmentación, se usa la Tabla 3.3; para la parte de validación de candidatos, se usa la Tabla 3.4 y para la obtención de las otras etiquetas que se necesitan, se usa la Tabla 3.11.

4. EVALUACIÓN DEL SISTEMA

En este capítulo se van a mostrar y analizar los resultados obtenidos a lo largo del proyecto, tanto para la parte de detección como para la parte de etiquetado. El objetivo principal del proyecto no es obtener un sistema óptimo, sino investigar los algoritmos y ver como optimizar el sistema para una base de datos determinada.

Para comenzar, se tienen que comparar los dos métodos usados en la fase de detección usando la base de datos A. Posteriormente se analizará de la etapa de etiquetado como código independiente, donde se detecta manualmente la cara con las imágenes de la base de datos B. Por último se unen ambas partes formando la totalidad del proyecto y se evaluará el conjunto mediante la base de datos A.

En el contexto de las tareas de clasificación, es necesario definir una serie de parámetros [20]:

- True positive o verdadero positivo, TP → aceptar una hipótesis positiva cuando es positiva.
- False negative o falso negativo, FN → rechazar una hipótesis positiva cuando en realidad es positiva.
- False positive o falso positivo, FP → aceptar una hipótesis positiva cuando en realidad no es positiva.
- True negative o verdadero negativo, TN → aceptar una hipótesis negativa cuando es negativa.

		VALOR PREDICHO	
		Positivo	Negativo
VALOR REAL	Positivo	TP	FP
	Negativo	FN	TN

Tabla 4.1: Matriz de confusión

- Precisión (precisión en ingles) → es el número de elementos positivos detectados (TP), entre el total de los etiquetados como positivos (lo sean o no). Si la precisión=1 significa que todos los objetos etiquetados como positivos lo son, pero puede haber positivos que no sean detectados.

$$p = TP / (TP + FP)$$

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- Cobertura (recall en ingles) → es el número de elementos positivos detectados (TP), entre el total de los que realmente pertenecen a la clase positiva (detectados o no). Si la cobertura=1 significa que se han detectado todos los elementos positivos, pero puede que también se haya etiquetado como positivo alguno que no lo es.

$$r = TP/(TP + FN)$$

Hay veces en las que se preferiría que la precisión sea mayor aunque la cobertura sea más baja, por ejemplo en control de acceso de una empresa, se puede preferir que todas las caras que detecte sean de empleados, aunque haya algunos empleados que no se detecten, así evitarían intrusiones. También puede darse el caso a la inversa, por ejemplo en sistemas de vigilancia, que se preferiría detectar todas las caras existentes aunque con ellas se seleccionen algunas que no lo son, para poder controlar de manera segura todas las que sí lo son y luego desechar el resto.

Una forma de llegar a un acuerdo entre ambas medidas es la media ponderada.

- Medida F (F-measure en ingles) → es la media ponderada de los armónicos de la precisión y la cobertura.

$$f1 = 2 \times [(p \times r)/(p + r)]$$

4.1 FASE DE DETECCIÓN

En el capítulo anterior se detalló la implementación del sistema para la detección de caras en imágenes mediante dos técnicas diferentes. Ambos métodos van a ser evaluados con la base de datos A, donde tenemos 50 fotografías en un ambiente no controlado.

Recordar que el primer método empleado para detectar caras está basado en características de bajo nivel, más concretamente en el espacio de color RGB, mientras que la segunda técnica se basa en un modelo basado en plantillas.

Se realizan varias pruebas cambiando las operaciones morfológicas, los umbrales de candidatos a caras, el valor de la correlación, etc. Mediante las medidas explicadas en el apartado 4, se analizará la respuesta de los algoritmos en función de las diferentes pruebas.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Para medir los errores y los aciertos del sistema, se siguen una serie de pautas:

- Se considera una cara como válida, cuando se recuadra la cara entera o cuando se marca la cara entera y además contiene parte de pelo o de cuello.
- No se considera una cara como válida, cuando no esté la cara entera recuadrada, cuando se marcan varias caras juntas como una sola o cuando se seleccione la cara junto al escote o al pecho.

4.1.1 SELECCIÓN DE LOS OPERADORES MORFOLÓGICOS

Como se mencionó en el capítulo 3, Diseño e Implementación, en los dos métodos de detección se usa tanto erosión como dilatación. En ambos casos, el tamaño del elemento estructurante es de vital importancia, y debido a la heterogeneidad de las imágenes de la base de datos, ha costado mucho su elección.

Para el método de *detección de caras de bajo nivel*, se probó tanto para valores iguales del tamaño del elemento estructurante en ambas operaciones, como para valores diferentes, siempre con forma de disco. En la tabla 4.2 se muestran los resultados para distintos tamaños del elemento estructurante, teniendo en cuenta que se tienen las 50 imágenes de la base de datos A con 76 caras.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>f1</u>
EE tamaño 3-3	47	29	44	0.516	0.618	0.562
EE tamaño 5-5	34	42	35	0.492	0.447	0.468
EE tamaño 3-5	52	24	121	0.3	0.684	0.4329

Tabla 4.2: Resultados operadores morfológicos para detección de bajo nivel

Como se puede observar para el caso tercero, en el que el tamaño del elemento estructurante de la erosión es 3 y el de la dilatación es 5, tenemos la mayor tasa de verdaderos positivos, TP, lo cual podría significar que es la mejor elección, pero esta medida no representa nada si no se compara también las otras. En este ejemplo es donde más caras se detectan correctamente pero esto viene promovido porque aquí, se detectan casi todo como caras, de ahí su alta tasa de falsos positivos.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Cuando se usa el elemento estructurante de tamaño 5, se ve que del total de caras (76), detecta prácticamente la mitad de las caras, aunque también es cierto que falla bastante poco.

Lo descrito aquí arriba es lo mismo que hablar en términos de precisión, cobertura y F-Measure. Si la precisión es muy alta (próxima a uno), significaría que todas las caras que detecta el algoritmo son correctas, pero no dice nada de las caras que no detecta. Por el contrario, si la cobertura esta cerca de uno, implicaría que detecta todas las caras existen, pero no especifica si además detecta otras que no lo son. Esto es aproximadamente lo que pasa para el tercer caso.

Para tomar la decisión nos fijamos en la F-measure, para que estén compensados los valores de precisión y cobertura. Como se puede observar, este parámetro da mejor para el primer caso, ya que detecta bastantes caras correctamente y a la vez no detecta demasiadas caras que no lo son en realidad. Por este motivo, se concluye que la mejor solución se da con el tamaño del elemento estructurante 3 para ambos operadores.

Para el método de **detección de caras basado en plantillas**, se prueba también con diversos tamaños del elemento estructurante, iguales o diferentes para los dos operadores usados. Los resultados que se obtienen se pueden ver en la Tabla 4.3.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>f1</u>
EE tamaño 4-4	47	29	51	0.479	0.618	0.539
EE tamaño 8-8	46	30	40	0.535	0.605	0.567
EE tamaño 10-8	47	29	29	0.618	0.618	0.618

Tabla 4.3: Resultados operadores morfológicos para detección con plantillas.

Como se puede comprobar pasa algo parecido al anterior detector; en el ejemplo primero, se tiene el mayor número de caras acertadas, pero también es porque detecta muchas caras que realmente no lo son. Esto implica un valor de cobertura bastante elevado, pero con una precisión muy baja. En el segundo, se detectan menos caras pero también es menor el valor de falsos negativos.

Se ha elegido el tercer caso, porque es donde mejor F-measure se tiene, ya que hay una buena compensación entre el número de caras detectadas y el número de veces que detecta cara cuando no la hay.

4.1.2 CAMBIO DE TAMAÑO DE IMAGEN DE ENTRADA

Al ir cambiando el elemento estructurante y viendo los resultados para las distintas fotos, se observó que cuanto más gente hubiera en una foto, al ser las caras más pequeñas, peor resultaba la detección. Esto viene derivado del tamaño del elemento estructurante, ya que las caras están tan juntas en la foto, que detecta todo como una "masa" de piel, por lo que luego no cumple las condiciones para ser cara.

Se decidió que era mejor usar un tamaño pequeño de foto, aunque se haga peor la detección en el caso en el que sale mucha gente, a cambio de reducir el tiempo de ejecución.

Para el caso de **detección de bajo nivel**, como se ve en la Figura 4.1, cuanto mayor es la foto, mejor detecta las caras. Hay que tener en cuenta, que en este caso no es muy eficaz el detector por culpa del elemento estructurante, pero como se vio en el apartado 4.1.1 se optimiza el tamaño del elemento en función de la heterogeneidad de la base de datos A.

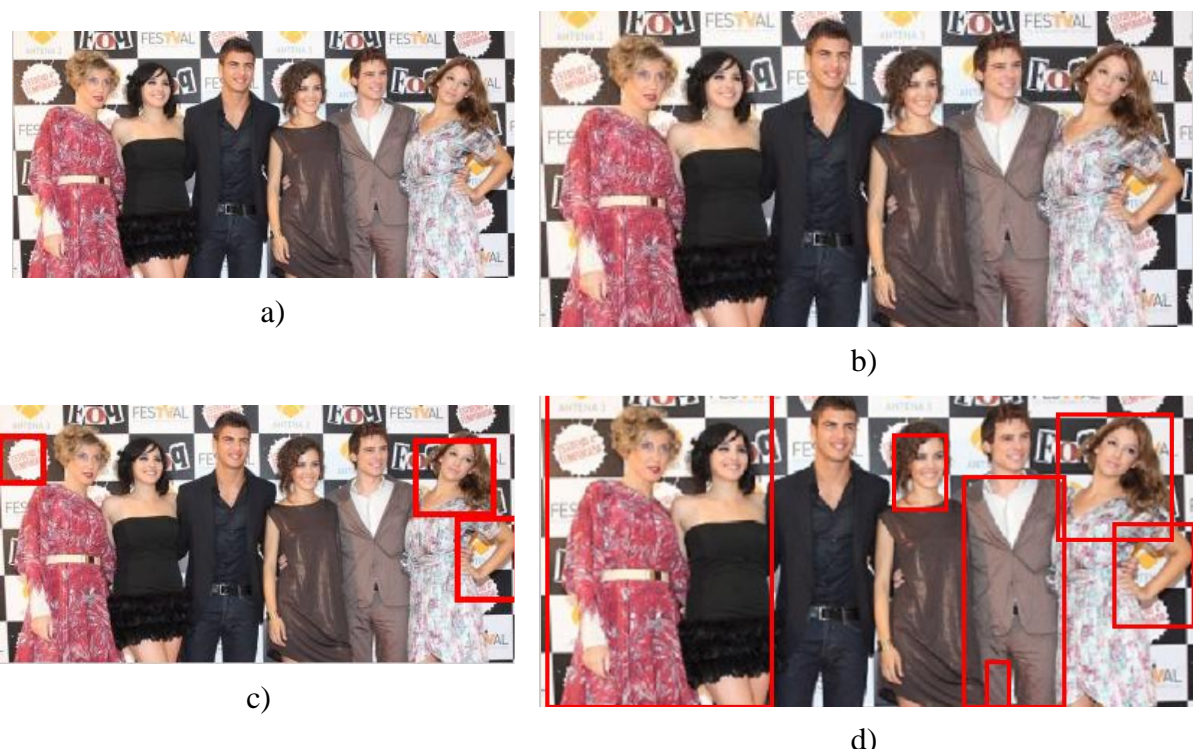


Figura 4.1: a) Imagen de la base de datos A con tamaño (351x172);
b) Imagen con tamaño (631x313); c) Caras detectadas de la imagen a);
d) Caras detectadas de la imagen b).

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Según los resultados de la Tabla 4.4, se obtiene una mejor precisión y cobertura, y por lo tanto, mejor media ponderada, con el tamaño de foto más grande, pero también se ve un gran aumento en el tiempo de ejecución. Como se comentó anteriormente, este es el motivo por el que se prefiere coger fotos de tamaño más reducido, sacrificando la correcta detección en imágenes con muchas caras.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>	<u>TIEMPO</u>
Foto tamaño (351x172)	1	5	2	0.333	0.166	0.221	1.38
Foto tamaño (631x313)	2	4	4	0.33	0.333	0.333	3.04

Tabla 4.4: Resultados del cambio de tamaño para la detección de caras de bajo nivel.

En el caso de **detección basado en plantillas**, se usan las mismas fotos con los mismos tamaños y se obtienen los siguientes resultados.



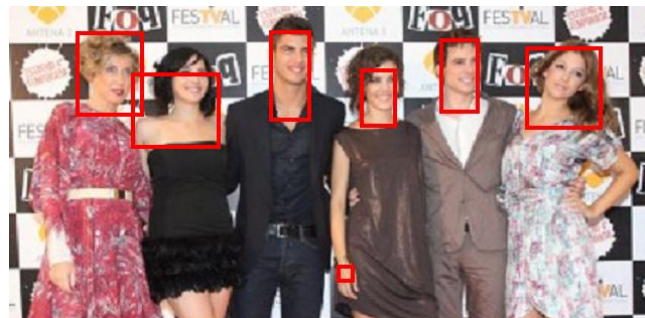
a)



b)



c)



d)

Figura 4.2: a) Imagen de la base de datos A con tamaño (351x172);
b) Imagen con tamaño (631x313); c) Caras detectadas de la imagen a);
d) Caras detectadas de la imagen b).

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>	<u>r</u>	<u>F1</u>	<u>TIEMPO</u>
Foto tamaño (351x172)	3	3	0	1	0.5	0.666	8.00
Foto tamaño (631x313)	5	1	2	0.714	0.833	0.769	19.98

Tabla 4.5: Resultados del cambio de tamaño para la detección de caras basado en plantillas.

Al igual que pasaba con el otro detector, al aumentar el tamaño de la foto detecta mejor las caras. Como se puede ver en la Tabla 4.5, la media ponderada es bastante alta pero el tiempo de procesamiento es más del doble que en la foto de tamaño menor, por lo que se decide que es mejor usar las imágenes más pequeñas.

4.1.3 CAMBIO DE UMBRALES DE DETECCIÓN

Las siguientes pruebas que pueden hacerse son sobre los umbrales que se establecen a la hora de decidir los candidatos a caras. Como se explicó en 3.5.1.3, existen una serie de parámetros en los que se basa la decisión.

En el **algoritmo de bajo nivel** se emplean el factor de forma, la relación de aspecto, la solidez y el número de agujeros. Con este último valor no se hicieron pruebas, ya que se considera que todas las caras deben tener al menos un agujero para considerarse cara, como se explica en el capítulo de Diseño e Implementación.

La primera propiedad con la que se probó es el factor de forma, que mide la regularidad de una región. Sabemos que si este parámetro es 1, significa que tendrá forma circular, así que al ser una cara parecida a una elipse, se dedujo que el resultado tenía que ser bastante menor.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>	<u>r</u>	<u>F1</u>
Factor forma > 0.1	47	29	44	0.516	0.618	0.562
0.1 < Factor forma < 0.3	15	61	35	0.3	0.197	0.237
Factor forma > 0.3	31	45	24	0.563	0.407	0.486

Tabla 4.6: Resultados de cambiar el factor de forma en la detección de bajo nivel

Como se puede ver en la Tabla 4.6, donde más caras se detectan, aun sin tener un número de falsos positivos muy alto, es para el factor de forma mayor que 0.1, que como vemos es la que nos da la media ponderada más elevada, por lo que se elegirá ese valor para el algoritmo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Otra propiedad que se modificó para ver como afectaba al código es la relación de aspecto, que define la relación entre el diámetro mayor y menor de la región. Se prueba para valores mínimos de 0.1, 0.3 y 0.6, y como se puede ver en la Tabla 4.7, la mejor media ponderada se da en el segundo caso estudiado, por lo que al final es el que se usa. Para la relación de aspecto mayor que 0.1, vemos que se obtienen muchas caras detectadas que son correctas, pero también muchas que no lo son. Cuando se pone el umbral en 0.6, se falla mucho menos pero también se acierta muy poco.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Relación aspecto>0.1</i>	46	30	65	0.414	0.605	0.491
<i>Relación aspecto>0.3</i>	47	29	44	0.516	0.618	0.562
<i>Relación aspecto>0.6</i>	31	45	23	0.574	0.407	0.476

Tabla 4.7: Resultados de cambiar al relación de aspecto en la detección de caras de bajo nivel.

Las últimas pruebas que se realizan para este método, se hacen en base a la solidez, que indica el porcentaje de la región que está dentro del cerco convexo. Si se usa para valores menores que 0.52, no se detecta nunca ninguna cara. Como se ve en la tabla 4.7, en ambos casos estudiados tenemos un valor de verdaderos positivos bastante elevado, pero en el caso en el que umbral es mayor de 0.7 se reduce el número de caras detectadas que realmente no lo son, por lo que su media ponderada es más elevada. Por este motivo se eligió el valor mínimo de 0.7 para la solidez.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Solidez>0.52</i>	48	28	55	0.466	0.631	0.537
<i>Solidez>0.7</i>	47	29	44	0.516	0.618	0.562

Tabla 4.8: Resultados de cambiar la solidez en la detección de bajo nivel.

En resumen, en el algoritmo de detección de rostros de bajo nivel, experimentalmente se llega a la conclusión de que los mejores resultados posibles se dan cuando:

- Factor de forma>0.1
- Relación de aspecto>0.3
- Solidez>0.7

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Ahora se va a realizar el mismo procedimiento con diferentes umbrales que se aplican en la **detección de caras basada en plantillas**.

En este método se emplean el número de agujeros, la relación de aspecto y la correlación entre la imagen y una plantilla robotizada. Al igual que pasaba con el otro algoritmo, el número de agujeros siempre tiene que ser mayor que uno, por lo que no se realizan pruebas al respecto.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
Relación aspecto>1	42	34	33	0.56	0.552	0.556
1<Relación aspecto<3.5	47	29	29	0.618	0.618	0.618
Relación aspecto>2	3	73	7	0.3	0.039	0.069

Tabla 4.9: Resultados de cambiar la relación de aspecto en la detección basada en plantillas.

En cuanto a la relación de aspecto, se tiene los resultados en la Tabla 4.9 y se comprueba que el mayor valor de la media ponderada se da cuando el ratio varía entre 1 y 3,5. En el caso en que se pone de umbral mínimo 1, se tiene una precisión y una cobertura bastante buenas, pero no tanto como en el caso anterior. Cuando el valor mínimo se establece en 2, no se detectan prácticamente caras, ni correctas ni incorrectas. Por esto, se decidió elegir la relación de aspecto entre 1 y 3.5.

La correlación es seguramente uno de los valores más importantes a la hora de la detección. Como se explicó en el capítulo 3, la correlación mide cuánto de parecidas son dos imágenes, siendo 1 para imágenes iguales. En este caso se usa una plantilla robotizada y como hay que tener en cuenta que la región candidata a cara puede estar girada, o recortada, se supone que el valor de correlación tendrá que ser algo menor que uno.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
Correlación>0.3	47	29	30	0.61	0.618	0.613
Correlación>0.6	47	29	29	0.618	0.618	0.618
Correlación>0.8	15	61	6	0.714	0.197	0.308

Tabla 4.10: Resultados de cambiar la correlación en la detección basada en plantillas.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Como se puede ver en la Tabla 4.10, para un valor mayor de 0.8, casi no se detectan caras, por lo que su media ponderada es muy baja. En cualquiera de los otros dos casos, se obtienen mejores resultados tanto de precisión como de cobertura, pero por ser algo mayor, se decide que el valor mínimo de correlación tiene que ser 0.6.

En resumen, en el algoritmo de detección de rostros basada en plantillas, experimentalmente se llega a la conclusión de que los mejores resultados posibles se dan cuando:

- $1 < \text{Relación de aspecto} < 3.5$
- $\text{Correlación} > 0.6$

4.1.4 RESUMEN DE LOS ALGORITMOS

Tras las diferentes pruebas, se eligieron los valores que dieron las mejores marcas posibles en cada caso. Se ejecutaron los dos detectores de caras con las 50 imágenes de la base de datos A, cuyos resultados pueden consultarse en el Anexo 6.1 y 6.2.

La Tabla 4.11 muestra una comparación entre el funcionamiento de los dos métodos para todas las imágenes. En ella se calculan la precisión, la cobertura, la media ponderada y además el tiempo medio de ejecución para cada algoritmo con las 50 imágenes. Hay que tener en cuenta que todos los algoritmos están probados con un ordenador AMD Sempron Processor 3400+ con 2.01GHz y 1.00G de RAM.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>	<u>Tiempo medio</u>
<i>Bajo nivel</i>	47	29	44	0.516	0.618	0.562	1.396
<i>Plantillas</i>	47	29	29	0.618	0.618	0.618	7.957

Tabla 4.11: Comparación entre los algoritmos de detección de caras.

Como puede observarse, la cobertura es la misma para ambos casos, es decir, los dos métodos detectan el mismo número de caras correctas de todas las existentes. Por el contrario, la detección basada en plantillas, tiene una precisión mayor, lo cual significa que el método de bajo nivel etiqueta más veces regiones que no son caras, como si fueran correctas.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Para un proyecto de etiquetado de imágenes, podría decirse que es más importante que haya una gran precisión, es decir que se falle lo menos posible, aunque con eso se detecten menos caras. La detección basada en plantillas, aparte de tener una mayor precisión, tiene una mejor media ponderada, $F1$, por lo que podría considerarse un método bastante bueno para nuestro sistema. El problema de este algoritmo en comparación con el otro, es su velocidad de procesamiento, por lo que si hubiera limitaciones de tiempo de ejecución, sería más recomendable elegir el método de bajo nivel.

Ahora se pasa a discriminar el análisis de los algoritmos en función de algunas características de las imágenes, para comparar, si existe, una mejora en función de esas cualidades. Es decir, del total de las imágenes, descubrir si existe algún motivo por el que se detectan unas caras correctamente y otras no. Estas características son: el número de personas que aparecen en cada imagen, la raza y el sexo de las personas.

4.1.4.1 Evaluación en función del n° de personas

Se trata de ver para cada método si alguno de ellos detecta mejor las caras en función del número de personas que aparecen en la foto. En la base de datos A, tenemos 28 imágenes donde solo sale una persona y 22 imágenes donde sale más de una persona, de las cuales: en 13 de ellas salen dos personas, en una salen tres personas, en dos imágenes salen cuatro personas, en una imagen salen cinco personas y en otra imagen salen seis personas.

En la **detección de bajo nivel** se obtienen los resultados de la Tabla 4.12. En ella se ve cómo para cada categoría se obtienen los valores de TP (n° de caras que detecta de todas las fotos de esa categoría), de FN (n° de caras que no detecta de esa categoría) y de FP (regiones que detecta como caras que no lo son).

Con ello se calcula la precisión, la cobertura y la media ponderada, y como puede observarse, estos valores son superiores para las imágenes de cuatro caras. Esto no podría generalizarse como que funciona mejor para esa categoría, sino que al pertenecer a ella tan solo dos imágenes, estas pueden tener otras características, por ejemplo un fondo uniforme y distinto al color de la piel, que faciliten su detección.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Una mejor comparación sería entre las imágenes que tienen una sola cara y aquellas que tienen más de una cara. Como se puede ver en la Tabla 4.12, cuando se tiene una imagen de una sola persona, su precisión es menor, es decir se falla mucho más, pero en cambio su cobertura es mayor, ósea, que detecta más caras correctas que cuando se tienen varias personas juntas en una sola imagen.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
1cara	20	8	30	0.4	0.714	0.5712
2caras	19	7	12	0.613	0.730	0.666
3 caras	2	1	1	0.666	0.666	0.666
4 caras	7	1	1	0.875	0.875	0.875
5 caras	3	2	0	1	0.6	0.75
6 caras	1	5	4	0.2	0.166	0.181
Más de una cara	32	16	18	0.64	0.666	0.652

Tabla 4.12: Evaluación en función del n° de personas, de la detección de bajo nivel.

En el *método basado en plantillas*, ocurre algo similar que en el caso anterior. La media ponderada más alta se da en la imagen donde aparecen cinco caras, pero no se puede asegurar que el método funcione mejor para estas imágenes. Sin embargo, aquí también tenemos mejor media ponderada para las imágenes con más de una cara, tanto en precisión como en cobertura.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
1cara	15	13	18	0.454	0.535	0.491
2caras	17	9	9	0.653	0.653	0.653
3 caras	2	1	0	0.666	1	0.799
4 caras	6	2	0	0.75	1	0.857
5 caras	4	1	0	0.8	1	0.888
6 caras	3	3	0	0.5	1	0.666
Más de una cara	32	16	9	0.78	0.666	0.718

Tabla 4.13: Evaluación en función del n° de personas, de la detección basado en plantillas.

4.1.4.2 Evaluación en función de la raza

Esta comparación está basada en las diferentes razas que aparecen en las imágenes de la base de datos A: asiática, negra, árabe y blanca. Hay que tener en cuenta, que la mayoría de las fotos son de personas españolas, por lo que habrá una raza predominante, y no se podrá generalizar en cuanto la eficiencia del detector según la raza.

En la **detección de bajo nivel**, como se puede observar en la Tabla 4.14, existe una categoría, la de los árabes, que tiene precisión, cobertura y media ponderada de valor 1. Esto se explica porque se tiene una única foto con dos personas y el detector localiza exacta y únicamente, las dos caras. Como se mencionó arriba, esto no vale para generalizar, por lo que es mejor comparar las imágenes con personas blancas y las que no.

Como tanto la raza asiática, como la negra, y la árabe, tienen unos buenos resultados, el conjunto de todos ellos, dará también buenos resultados, siendo mejor la detección para esta categoría, que para personas blancas.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
Asiáticos	5	1	4	0.555	0.833	0.666
Negros	3	1	2	0.6	0.75	0.666
Árabes	2	0	0	1	1	1
Blancos	42	22	42	0.5	0.656	0.567
No blancos	10	2	6	0.625	0.833	0.714

Tabla 4.14: Evaluación en función de la raza, de la detección de bajo nivel.

Para el caso de de la **detección de rostros basada en plantillas**, cuyos resultados aparecen en la Tabla 4.15, vuelve a salir un valor alto de la media ponderada en la categoría árabe. Esto nos lleva a pensar, que esta imagen, tiene otras características que facilitan la detección.

En cuanto a las imágenes con personas de raza blanca y las imágenes que no son de raza blanca, tienen un valor parecido de media ponderada, provocada porque uno tiene un mayor valor de precisión y el otro de cobertura.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Asiáticos</i>	3	3	0	1	0.5	0.666
<i>Negros</i>	1	3	0	1	0.25	0.4
<i>Árabes</i>	2	0	1	0.666	1	0.799
<i>Blancos</i>	41	23	28	0.594	0.64	0.616
<i>No blancos</i>	6	6	1	0.857	0.5	0.631

Tabla 4.15: Evaluación en función de la raza, de la detección basada en plantillas.

4.1.4.3 Evaluación en función del sexo

Por último, se analizará la respuesta del sistema en función del sexo. Para este caso se cogen solo las 28 imágenes de la base de datos A donde aparece una única persona. Existen 13 imágenes de mujeres, 13 imágenes de hombres y dos imágenes donde salen niños.

En la **detección de bajo nivel**, se evalúan estas imágenes y se obtienen los valores que nos hablarán de su eficiencia. Como se ve en la Tabla 4.16, al resultado más alto de la media ponderada corresponde a las imágenes de mujeres, ya que en estos casos falla menos al etiquetar caras incorrectas.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Mujeres</i>	9	4	11	0.45	0.692	0.545
<i>Hombres</i>	10	3	15	0.4	0.769	0.526
<i>Niños</i>	1	1	4	0.2	0.5	0.285

Tabla 4.16: Evaluación en función del sexo, en la detección de bajo nivel.

En el algoritmo de **detección basado en plantillas**, por el contrario, se obtiene un valor más alto de la media ponderada en el caso de los niños, ya que detecta correctamente todas las caras que hay, aunque además etiquete como caras dos que no lo son.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
Mujeres	6	7	8	0.428	0.461	0.443
Hombres	7	6	8	0.466	0.538	0.499
Niños	2	0	2	1	0.5	0.666

Tabla 4.17: Evaluación en función del sexo, en la detección basada en plantillas.

4.2 FASE DE ETIQUETADO

En este apartado se va a analizar los resultados obtenidos por la fase de etiquetado, de manera independiente a todo lo que se ha visto anteriormente.

En este caso se tiene un conjunto de 40 imágenes de la base de datos B, donde se ha seleccionado la cara manualmente y se le pasa al algoritmo estas coordenadas. Se van a realizar pruebas con diferentes umbrales y restricciones que existen en este método, para explicar el porqué de su elección.

Por último, con el algoritmo ya definitivo, se calculan las medidas explicadas en el principio de este capítulo, que son: TP, FN, FP, p, r y F1.

Hay que tener en cuenta que el etiquetado de las diferentes categorías, es algo bastante subjetivo, por lo que la manera de medir los aciertos en cada caso, podría variar en función de la persona que lo evalúe. Por ejemplo, si tenemos una persona con el pelo castaño, podría darse el caso de que hubiera gente que la etiquetara como morena, si fuera castaña oscura y gente que la podría etiquetar como rubia, si fuera castaña clara. Además hay que saber que influye la luz en el tono de las cosas (tanto para el color de pelo, como para barba, raza, etc.), por lo que una misma persona podría verse con diferente tonalidad en distintas imágenes.

Como ya no se tiene una decisión binaria como antes (1 si es cara, 0 si no es cara), sino que se tienen varias categorías (por ejemplo, en el color del pelo: moreno, pelirrojo, castaño y rubio), se usará una matriz de confusión. Esta es una herramienta de visualización que se emplea en el aprendizaje supervisado y sirve para ver si el sistema se confunde entre varias clases. Cada columna de la matriz de confusión representa el número de precisiones de cada clase, mientras que cada fila representa a las instancias en la clase real [16].

4.2.1 CAMBIO DE UMBRALES

Las etiquetas que se van a obtener para cada imagen son: raza, color de ojos, color de pelo, barba, longitud del pelo y sexo.

Para etiquetar la raza, el color del pelo, la barba y la longitud del pelo, se utilizan solo umbrales obtenidos mediante el entrenamiento con otras imágenes o con histogramas, por lo que no se pueden hacer pruebas sobre ellos. Por el contrario en el etiquetado de ojos y el sexo si pueden variar algunos parámetros como son, el factor de forma, la relación de aspecto, etc.

En el etiquetado del color de ojos, como ya se explicó en el apartado 3.2.6, se segmenta basándose en el espacio de color de RGB, se calcula la inversa, ya que los ojos que es lo que se quiere detectar, no son piel y se obtiene el factor de forma y el área de cada región de "no piel". Se prueba con diferentes valores para el factor de forma y con un umbral distinto para el área, y se mide cuando se detectan los ojos correctamente.

4.2.1.1 Factor de forma en etiquetado de ojos

Se prueba a cambiar el valor del factor de forma, que es la propiedad que mide la regularidad de la región, y al ser ésta un ojo, tiene una forma parecida a una elipse.

Como puede verse en la Tabla 4.18, se pasan las 40 imágenes de la base de datos A, con diferentes umbrales del factor de forma, y se mira cuántos ojos se detectan correctamente, cuántos se detectan incorrectamente y cuántos no se detectan.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Factor forma>0.1</i>	57	23	178	0.242	0.712	0.361
<i>Factor forma>0.2</i>	51	29	141	0.265	0.637	0.374
<i>Factor forma>0.4</i>	27	53	82	0.247	0.337	0.285

Tabla 4.18: Pruebas en el factor de forma.

En el primer caso, el problema está en que detecta muchas caras correctamente pero sacrificando así las regiones que no son caras y las detecta como tal.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Cuando el factor de forma es mayor que 0.2 vemos que se da la media ponderada más alta, ya que se detectan bastantes caras correctas sin tener un número muy elevado de falsos positivos. Al poner el factor de forma mayor que 0.4, se disminuyen notablemente las caras que se etiquetan como tal, tanto las correctas como las fallidas. Por este motivo, se ha elegido el umbral de 0.2

4.2.1.2 Tamaño del área en etiquetado de ojos

La otra propiedad para detectar ojos, es el tamaño del área. Como tenemos imágenes de diferente resolución y además las caras que aparecen en ellas unas veces son más grandes que otras, no se podía fijar el umbral para el área de manera fija, sino variante. Por este motivo, se prueban varios umbrales en función del ancho de la cara que se pasa manualmente.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
$w/2$	38	42	110	0.256	0.475	0.332
$w/3$	51	29	141	0.265	0.637	0.374

Tabla 4.19: Prueba de tamaño del área.

Como se ve en la Tabla 4.19, los mejores resultados se dan cuando el umbral es mayor que un tercio del ancho de la cara, w . Además para este caso es mejor detectar correctamente el mayor número posible de ojos, aunque con ello se equivoque el sistema ligeramente más. Por este motivo, se elige como umbral para el tamaño de área $w/3$.

4.2.1.3 Factor de forma en etiquetado de sexo

Como se explicó en el apartado 3.6.5, para etiquetar el sexo de una persona, se tomaban varios parámetros: ser calvo, tener barba, un factor de sexo y tener el pelo largo. Todos ellos se obtienen de los otros métodos para el etiquetado, excepto el factor de sexo. Este es un valor inventado, en el que se tienen en cuenta el factor de forma y la relación de aspecto de una cara, para determinar si es un hombre o una mujer.

Se hacen dos pruebas, la primera siendo el factor de forma superior a 0.1, y la segunda siendo el factor de forma entre 0 y 0.2. Cuando cumple esta condición, más la de la relación de aspecto menor que 0.7, se etiqueta como mujer. Los resultados pueden verse en la Tabla 4.20.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Factor forma>0.1</i>	23	17	17	0.575	0.575	0.575
<i>Factor forma<0.2</i>	24	16	16	0.6	0.6	0.6

Tabla 4.20: Prueba del factor de forma en etiquetado de sexo.

Según esta tabla puede verse que se obtienen un poco mejores los resultados cuando el factor de forma está entre 0 y 0.2.

4.2.1.4 Relación de aspecto en etiquetado de sexo

Por último nos queda hacer pruebas con el ratio, que es la propiedad que mide la relación entre el diámetro mayor y el menor de una región.

Se considera que una mujer tendrá menor relación de aspecto, ya que tiene la cara más estrecha, por lo que la diferencia entre sus ejes será más grande. Para que el factor de sexo nos diga que la cara es una mujer, se prueba con dos valores de relación de aspecto, como se ve en la Tabla 4.21.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>	<u>r</u>	<u>F1</u>
<i>Relación de aspecto<0.6</i>	23	17	1	0.575	0.575	0.757
<i>Relación de aspecto<0.7</i>	24	16	16	0.6	0.6	0.6

Tabla 4.21: Prueba de la relación de aspecto en etiquetado de sexo.

Para el caso en el que se tiene una relación de aspecto menor que 0.7, se obtiene una mayor precisión, cobertura y media ponderada, por tanto es el valor que usaremos para el algoritmo final.

4.2.2 RESUMEN DEL ALGORITMO DE ETIQUETADO

Tras las diferentes pruebas que se han realizado, se han elegido los valores que han dado mejores resultados, y con ellos se ha creado el algoritmo de etiquetado final.

Como ya se mencionó, se usan las 40 imágenes de la base de datos B, se ejecuta el algoritmo y se realizan diferentes test en función de las características que queramos etiquetar.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Primero se evalúa cada etiqueta por separado, para ver sus prestaciones de manera individual y si existe confusión entre unas categorías y otras.

Posteriormente se realiza un resumen del conjunto de todas las etiquetas, para comprobar cuál de ellas es la más acertada y también se analizarán que imágenes de la base de datos B obtienen el mayor número de etiquetas correctas.

4.2.2.1 Raza

El algoritmo, como se explicó en el apartado 3.6.1, es capaz de detectar personas de piel negra y blanca. Dentro de las personas de piel negra, se pueden encontrar imágenes de árabes, los cuales generalmente suelen tener un color de piel bastante oscuro, mientras que los asiáticos suelen etiquetarse como blancos debido a su tono claro.

Se usa una matriz de confusión cuyas categorías de entrada están dispuestas en el eje "y" y cuyos valores predichos o etiquetas se sitúan en el eje "x", como muestra la Tabla 4.22.

		VALOR PREDICHO	
		<u>BLANCO</u>	<u>NEGRO</u>
VALOR REAL	<u>BLANCO</u>	30	6
	<u>NEGRO</u>	1	3

Tabla 4.22: Matriz de confusión del etiquetado de Raza.

Gracias a la Tabla 4.22, se ve claramente la relación entre categorías, ya que no solo es interesante saber cuándo acierta o falla una raza, sino que es importante saber que categorías se distinguen peor entre sí.

Como puede verse, en el caso de personas blancas, se etiquetan casi el 85% de las personas como blancas, mientras que solo se falla 6 veces de cada 36, etiquetándolo como raza negra. Cuando se tienen personas negras, se acierta un 75% mientras que el otro 25%, se etiqueta como si fueran personas blancas.

En el caso asiático y árabe, como ya se mencionó, hay que ver cada imagen puntualmente. Por ejemplo, en la imagen de test "cara39.jpg", que se ve en la Figura 4.3 a), se tiene a una mujer asiática con la piel clara, por lo que se debería de etiquetar como "Blanco". Si se mira en el anexo 6.3, se puede comprobar que es así, por lo que se contará como un acierto.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En la Figura 4.3 b), se tiene a una persona árabe de piel también clara, y si se mira en el anexo 6.3, se puede ver que lo etiqueta como "Blanco", por lo que se considerará un acierto.



a)



b)

Figura 4.3: Imágenes de la base de datos B; a) Imagen "cara39.jpg"; b) Imagen "cara40.jpg".

Los resultados de la evaluación en términos de precisión se presentan en la Tabla 4.23.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Blanco</i>	30	6	6	0.833
<i>Negro</i>	3	1	1	0.75

Tabla 4.23: Evaluación para el etiquetado de raza.

4.2.2.2 Color de ojos

De igual manera que con la raza, ahora se pasa a evaluar el etiquetado en función del color de ojos. Se crean las etiquetas: negros, marrones, verdes, azules y problema para etiquetar. Esta última categoría, se da en dos de las imágenes, ya que las personas aparecen con los ojos cerrados.

	<u>NEGROS</u>	<u>MARRONES</u>	<u>VERDES</u>	<u>AZULES</u>	<u>MAL</u> <u>ETIQUETADO</u>
<u>NEGROS</u>	7	0	1	4	3
<u>MARRONES</u>	0	9	1	3	2
<u>VERDES</u>	0	0	5	1	0
<u>AZULES</u>	0	0	0	2	0
<u>PROB. ETIQUETADO</u>	0	0	0	1	1

Tabla 4.24: Matriz de confusión del etiquetado del color de ojos.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Hay que tener en cuenta, como se mencionó al principio de este apartado, que la forma de evaluar si una etiqueta es cierta o no, se hace de manera subjetiva, de modo que una persona podría etiquetar a alguien con ojos verdes y otra persona diferente, podría decir que los tiene azules.

Como se muestra en la Tabla 4.24, cuando se tienen personas de ojos azules, se etiquetan de forma muy acertada, al igual que con los ojos verdes. En cambio se ve, que cuando son personas de ojos negros, no distingue muy bien esta categoría de la de ojos azules o de la etiqueta "mal etiquetado", que es cuando no detecta el ojo correctamente o tiene dudas de su color. Para que una mayor comprensión de los datos, se tiene la Tabla 4.25.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
Negros	7	8	8	0.466
Marrones	9	6	6	0.6
Verdes	5	1	1	0.833
Azules	2	0	0	1
Prob. etiquetado	1	1	1	0.5

Tabla 4.25: Evaluación para el etiquetado de color de ojos.

La precisión en el caso de ojos verdes y azules, como se adelantó anteriormente, es muy alta, lo cual es lógico debido a que la implementación estaba basada en el espacio de color RGB, siendo G y B, la componente verde y azul. Para el caso de ojos negros, se falla más veces de las que se acierta, lo cual viene derivado también del espacio de color RGB, ya que al ser el negro la ausencia de todos los colores, y al hacer la correlación por componentes, es más fácil, como pasaba en el caso anterior, que dé una mayor correlación para la componente verde o azul.

4.2.2.3 Color de pelo

Con este algoritmo se etiqueta el color del pelo de cada persona para las 40 imágenes, y una vez más, se hace la evaluación de manera subjetiva, debido a los cambios que se pueden producir por la luminosidad de la foto, los tintes que pueda llevar una persona en el pelo, etc.

Se crea una matriz de confusión en función a las categorías que son: moreno, castaño, rubio, pelirrojo y calvo. Se tienen las mismas etiquetas que categorías, como puede comprobarse en la Tabla 4.26.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>MORENO</u>	<u>CASTAÑO</u>	<u>RUBIO</u>	<u>PELIRROJO</u>	<u>CALVO</u>
<u>MORENO</u>	23	0	0	0	1
<u>CASTAÑO</u>	0	8	0	1	0
<u>RUBIO</u>	1	0	1	1	1
<u>PELIRROJO</u>	0	0	0	1	0
<u>CALVO</u>	0	0	0	0	2

Tabla 4.26: Matriz de confusión para el etiquetado del color de pelo.

Se ve que para los valores reales del color de pelo de las imágenes, no se produce mucha confusión entre las diferentes categorías. El peor caso que se obtiene es para las personas rubias, que como se puede comprobar en la Tabla 4.26, para una etiqueta correcta, falla otras cuatro. Lo bueno es que no se observa que haya dos categorías que se mezclen demasiado entre sí, ya que aunque en el caso de rubios se falla bastante, se hace de igual manera con las diferentes categorías.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Moreno</i>	23	1	1	0.958
<i>Castaño</i>	8	1	1	0.888
<i>Rubio</i>	1	3	3	0.25
<i>Pelirrojo</i>	1	0	0	1
<i>Calvo</i>	2	0	0	1

Tabla 4.27: Evaluación para el etiquetado del color de pelo.

Según la Tabla 4.27, se obtienen unos valores realmente buenos para el etiquetado de color de pelo. Se tienen que destacar de manera especial, el caso de las personas morenas, ya que son la mayoría de las imágenes y se detecta un 95% del total de los morenos.

4.2.2.4 Barba

Ahora se pasa a etiquetar la presencia o ausencia de barba en las imágenes de la base de datos B, que como ya se mencionó, es de vital importancia para etiquetar posteriormente el sexo de la persona. Se tienen dos etiquetas: con barba y sin barba, y su matriz de confusión se representa en la Tabla 4.28.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>BARBA</u>	<u>SIN BARBA</u>
<u>BARBA</u>	3	2
<u>SIN BARBA</u>	8	27

Tabla 4.28: Matriz de confusión para el etiquetado de barba.

Como se ve en la Tabla 4.28 se puede distinguir fácilmente entre personas con barba y sin ella, aunque cuando hay personas con barba existe mayor probabilidad de que se etiqueten de manera incorrecta.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
<i>Barba</i>	3	2	2	0.6
<i>Sin barba</i>	27	8	8	0.771

Tabla 4.29: Evaluación para el etiquetado de barba.

Si miramos la evaluación del conjunto, Tabla 4.29, cuando se tienen personas sin barba, se etiquetan más del 77% bien, mientras que para las personas con barba se etiquetan correctamente solo un 60%.

4.2.2.5 Longitud del pelo

Con este algoritmo se comprueba la longitud del pelo, que es otra de las características que se usarán para el etiquetado de sexo. En este caso también tenemos una decisión binaria.

	<u>LARGO</u>	<u>CORTO</u>
<u>LARGO</u>	11	4
<u>CORTO</u>	4	21

Tabla 4.30: Matriz de confusión para el etiquetado de la longitud del pelo.

Según la Tabla 4.30, se fallan el mismo número de veces tanto en personas con pelo corto como personas con pelo largo, aunque al ser mayor el número de personas de pelo corto, significa que es más preciso en este caso. A pesar de ello, se ve que no hay mucha confusión entre ambas clases.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
Largo	11	4	4	0.733
Corto	21	4	4	0.84

Tabla 4.31: Evaluación para el etiquetado de la longitud del pelo.

Como se puede comprobar, y ya se dijo anteriormente, la precisión en personas de pelo largo es bastante elevada y para personas de pelo corto es muy alta, por lo que esta medida será bastante acertada para luego utilizarla en el etiquetado de sexo.

4.2.2.6 Sexo

Por último está la etiqueta del sexo, que como se explicó en el apartado 3.6.6, es una característica que depende de otras, por lo que sus resultados están condicionados a los valores que se obtienen en las otras etiquetas.

También es una decisión binaria como en los apartados anteriores, pero al depender de muchos factores para su etiquetado, se crea una etiqueta de "no detectado" para aquellas imágenes cuyas respuestas de las otras propiedades (barba, calvo, longitud de pelo y factor de sexo) sean incongruentes entre sí.

Los resultados obtenidos al aplicar este algoritmo en la base de datos B, se muestran en la Tabla 4.32.

	<u>HOMBRE</u>	<u>MUJER</u>	<u>NO DETECTADO</u>
<u>HOMBRE</u>	17	4	0
<u>MUJER</u>	9	8	2

Tabla 4.32: Matriz de confusión para el etiquetado se sexo.

Según la Tabla 4.32, en la categoría de hombre se etiquetan casi todos correctamente, mientras que para las imágenes de mujeres se falla bastante más, es decir, la categoría de mujeres está un poco difusa respecto a la de hombre.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
Hombre	17	4	4	0.809
Mujer	8	11	11	0.421

Tabla 4.33: Evaluación para el etiquetado de sexo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En la Tabla 4.33 se muestra la evaluación del conjunto de imágenes para cada categoría. Se puede comprobar que la precisión en el caso de los hombres es muy alta, ya que se etiquetan correctamente algo más del 80% mientras que en el caso de las mujeres es prácticamente la mitad. Esto viene derivado de que las propiedades que componen el etiquetado de sexo, son más relevantes para etiquetar hombre. Por ejemplo, la presencia de barba, o la calvicie, implica que indudablemente es un hombre, pero su ausencia no implica que sea mujer.

Por este motivo, teniendo en cuenta que alguna de esas etiquetas pueden ser incorrectas, si varias de ellas son tal que se podría decir que es un hombre, seguramente lo sea. Si nuestras etiquetas nos dicen que no es calvo, no tiene barba y tiene el pelo largo, no podría asegurarse que por ello sea una mujer, por lo que la Tabla de la verdad representada en la Tabla 3.9, es más restrictiva para el caso de las mujeres.

4.2.2.7 Resumen del conjunto de etiquetado

Tras la evaluación de cada etiqueta por separado, se pasa ahora a obtener el resumen del conjunto de etiquetado.

Primero se analizará cada una de características que se etiquetan en una imagen, pero a diferencia de los apartados anteriores, se mirará la precisión general de cada etiqueta y no por categorías.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Raza</i>	33	7	7	0.825
<i>Color ojos</i>	24	16	16	0.6
<i>Color Pelo</i>	35	5	5	0.875
<i>Barba</i>	30	10	10	0.75
<i>Longitud de pelo</i>	32	8	8	0.8
<i>Sexo</i>	25	15	15	0.625
Precisión media				0.745

Tabla 4.34: Evaluación para el conjunto de etiquetado.

Según la Tabla 4.34, basándonos en las tablas por categorías de los apartados anteriores, se obtiene la precisión de cada etiqueta. Como se puede comprobar, la mayor precisión se da en el caso del color de pelo y la longitud del pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Los peores resultados aparecen en el color de ojos, lo cual es normal ya que en muchas fotos no se puede apreciar fácilmente el color, y en el etiquetado de sexo, que como se mencionó anteriormente, al ser una decisión tomada en base a las otras etiquetas, se van acumulando los errores obtenidos.

Ahora se pasa a evaluar cómo responde este sistema en función de la totalidad de las etiquetas que se obtienen. Es decir, se mirará la respuesta de cada imagen a las seis etiquetas estudiadas.

Los datos obtenidos sobre las 40 imágenes se muestran en el anexo 6.3, donde se señalan: sobre un fondo amarillo los resultados que tengan todas las etiquetadas marcadas correctamente, sobre un fondo azul las imágenes que tengan cinco de las seis etiquetas correctas, sobre un fondo verde las imágenes que tengan cuatro etiquetas ciertas y sobre un fondo amarillo las imágenes que tengan solo tres etiquetas correctas.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>6 etiquetas correctas</i>	8	32	32	0.2
<i>5 etiquetas correctas</i>	14	26	26	0.35
<i>4 etiquetas correctas</i>	10	30	30	0.25
<i>3 etiquetas correctas</i>	5	35	35	0.125
<i>0,1 o 2 etiquetas correctas</i>	3	37	37	0.075

Tabla 4.35: Evaluación de etiquetado.

Como se puede ver en la Tabla 4.35, de la totalidad de las imágenes, solo 8 de ellas obtienen las seis etiquetas de forma correcta, 13 imágenes detectan cinco de las seis etiquetas bien, 10 imágenes detectan cuatro ciertas, 5 imágenes detectan sólo 3 etiquetas correctas y las otras tres imágenes que quedan, detectan 0, 1 o 2 etiquetas bien. Con esto se comprueba que el caso que más se da, es acertar 5 etiquetas.

Además se tiene que observar con las imágenes que tienen más de cuatro etiquetas correctas, es decir, las 8 imágenes de seis etiquetas correctas, más las 14 imágenes de cinco etiquetas correctas, más las 10 imágenes de las cuatro etiquetas correctas, hacen el 80% de todos los datos, lo que nos lleva a pensar, que la mayoría de los datos tienen un nivel aceptable de etiquetado.

4.3 FASE FINAL: EVALUACIÓN DEL SISTEMA GLOBAL

Una vez se han analizado las fases de detección y de etiquetado por separado, se pasa a la fase final del proyecto, donde se evaluará el conjunto total para las imágenes de la base de datos A.

Como se tienen dos detectores de cara diferentes, primero se evaluará la detección de bajo nivel junto al etiquetado y posteriormente la detección basada en plantillas junto al etiquetado.

Hay que tener en cuenta, que ahora a la fase de etiquetado, no se le pasarán las coordenadas de la cara seleccionada manualmente, sino que será el resultado obtenido de la fase de detección. Por este motivo, puede haber caras recuadradas incorrectamente, lo que producirá un mal etiquetado.

4.3.1 DETECCIÓN DE BAJO NIVEL Y ETIQUETADO

Lo primero que se tiene que hacer es combinar el algoritmo de detección de bajo nivel basado en el espacio de color RGB con el algoritmo de etiquetado que se explicó en el capítulo 3, Diseño e implementación. Se ejecutará este código para las 50 imágenes de la base de datos A, obteniendo la cara, si la hay, recuadrada y utilizando sus coordenadas para etiquetar dicha cara.

Como ya se ha evaluado el algoritmo de detección de bajo nivel con la base de datos A en el apartado 4.1, se sabe que de las 76 caras que hay repartidas a lo largo de toda la base de datos, este método detecta 47 correctamente.

El análisis del sistema completo, se realiza evaluando la eficiencia del etiquetado para las caras detectadas, las cuales ya fueron evaluados en el algoritmo de detección de caras. Hay que resaltar que las coordenadas de estas caras, al no ser extraídas de manera manual, van a suponer un empeoramiento en la fase de etiquetado, al seleccionar, por ejemplo, caras con el pelo incluido.

Ahora se pasa a analizar sobre las 47 caras detectadas, cada una de las etiquetas de manera independiente, para ver su precisión y si existe confusión entre las distintas categorías.

4.3.1.1 Raza

La manera de evaluar el etiquetado de la raza se hace igual que cuando se tenía solo el algoritmo de etiquetado, pero ahora usando las coordenadas de la cara detectada por el detector de bajo nivel.

Se crea la matriz de confusión, cuyos valores predichos se sitúan en el eje "x" y cuyas categorías de entrada están dispuesta en el eje "y", como se ve en la Tabla 4.36.

		VALOR PREDICHO	
		<u>BLANCO</u>	<u>NEGRO</u>
VALOR REAL	<u>BLANCO</u>	22	21
	<u>NEGRO</u>	1	3

Tabla 4.36: Matriz de confusión para el etiquetado de raza.

Como se puede comprobar, entre personas blancas y de color existen unos valores bastantes altos, lo que indica que hay confusión entre ellos. Esto puede deberse a que si las coordenadas de la imagen no están debidamente ajustadas a la cara, esto puede provocar que el lugar de la frente donde se mira el color para obtener la raza, no sea tal lugar, por lo que los umbrales de la luminancia se verán modificados.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
<i>Blanco</i>	22	21	21	0.511
<i>Negro</i>	3	1	1	0.75

Tabla 4.37: Evaluación del etiquetado de raza.

Si se habla en términos de precisión, como se ve en la Tabla 4.37, un 75% de las personas negras se etiquetan correctamente mientras que solo un 51% de las blancas son correctas. Estos valores son más bajos que en la evaluación del algoritmo por separado (véase Tabla 4.22), debido a la imprecisión de las coordenadas y a que las imágenes de la base de datos A pueden estar giradas o muy alejadas del objetivo, produciendo peores resultados que con las imágenes de la base de datos B.

4.3.1.2 Color de ojos

La siguiente etiqueta a analizar es el color de ojos, que como ya se comentó, tiene diferentes etiquetas: negros, marrones, verdes, azules y problema para etiquetar. Esta última categoría, se da en casos, en los que la persona sale con los ojos cerrados, o prácticamente cerrados, impidiendo por tanto su etiquetado.

Hay que tener en cuenta, que la forma de analizar si una etiqueta es cierta o no, se hace de manera subjetiva, dependiendo de la persona que evalúa el sistema.

	<u>NEGROS</u>	<u>MARRONES</u>	<u>VERDES</u>	<u>AZULES</u>	<u>MAL ETIQUETADO</u>
<u>NEGROS</u>	4	0	5	7	1
<u>MARRONES</u>	0	7	4	7	1
<u>VERDES</u>	0	0	4	0	0
<u>AZULES</u>	0	0	0	3	0
<u>PROB. ETIQUETADO</u>	0	0	2	0	1

Tabla 4.38: Matriz de confusión del etiquetado de color de ojos.

Como se muestra en la Tabla 4.38, cuando se tienen personas de ojos azules, se etiquetan de forma muy acertada, al igual que con los ojos verdes. En cambio se ve, que cuando son personas de ojos negros o marrones, no distingue muy bien esta categoría de la de ojos azules o de ojos verdes. Para que una mayor comprensión de los datos, se tiene la Tabla 4.39.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Negros</i>	4	14	14	0.222
<i>Marrones</i>	7	12	12	0.368
<i>Verdes</i>	4	0	0	1
<i>Azules</i>	3	0	0	1
<i>Prob. etiquetado</i>	1	2	2	0.33

Tabla 4.39: Evaluación del etiquetado de color de ojos.

La precisión en el caso de ojos verdes y azules, como se adelantó anteriormente, es muy alta, lo cual es lógico debido a que la implementación estaba basada en el espacio de color RGB, siendo G y B, la componente verde y azul.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Para el caso de ojos negros, se falla más veces de las que se acierta, lo cual viene derivado también del espacio de color RGB, ya que al ser el negro la ausencia de todos los colores, y al hacer la correlación por componentes, es más fácil, como pasaba en el caso anterior, que me dé una mayor correlación para la componente verde o azul.

De nuevo, si se compara esta tabla con la Tabla 4.25, donde se evaluaba solo el algoritmo de etiquetado, se ve que disminuye la precisión para las categorías de ojos negros y marrones.

4.3.1.3 Color de pelo

Este algoritmo determina el color de pelo de las 47 caras detectadas con el método de bajo nivel. Se crean las siguientes etiquetas: moreno, castaño, rubio, pelirrojo y calvo.

Existen unas imágenes con las que hemos tenido cierto problema de etiquetado, estas son las personas con el pelo cano. Como no hay una etiquetado de "canoso", se ha decidido etiquetar como valido aquel valor al que se asemeje más el pelo. Por ejemplo, en la imagen 32 de la base de datos A, se tiene a una persona de pelo cano, pero se puede ver que existen restos de color moreno, por lo que se debería etiquetar como "Moreno". Como se ve en la Figura 4.4, la etiqueta que se obtiene es moreno, por lo que se considerará un acierto.



a)

Color de pelo: *Moreno*

b)

Figura 4.4: a) Imagen "32.jpg" de la base de datos A;
b) Resultado del etiquetado de color de pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

		VALOR PREDICHO				
		<u>MORENO</u>	<u>CASTAÑO</u>	<u>RUBIO</u>	<u>PELIRROJO</u>	<u>CALVO</u>
VALOR REAL	<u>MORENO</u>	34	0	0	1	3
	<u>CASTAÑO</u>	3	1	0	0	2
	<u>RUBIO</u>	1	0	0	1	1
	<u>PELIRROJO</u>	0	0	0	0	0
	<u>CALVO</u>	0	0	0	0	0

Tabla 4.40: Matriz de confusión del etiquetado de color de pelo.

En la Tabla 4.40 se representa la matriz de confusión, donde puede verse que la categoría de morenos tiene muy buenos resultados. Lo que más llama la atención de estos datos, es que se etiquetan varias personas como calvas cuando no lo son, y esto viene motivado por dos conceptos:

1. Que el recuadro de la cara no sea todo lo preciso que debería, por lo que la cara cogería también el pelo, y al mirar la zona donde supuestamente debería estar el pelo, se encuentra fondo y decide etiquetarlo como calvo.
2. Algunas de las fotos están tomadas en diferentes ángulos, por lo que la zona de pelo no está donde debería.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
<i>Moreno</i>	34	4	4	0.894
<i>Castano</i>	1	5	5	0.166
<i>Rubio</i>	0	3	3	0
<i>Pelirrojo</i>	0	0	0	0
<i>Calvo</i>	0	0	0	0

Tabla 4.41: Evaluación del etiquetado del color de pelo.

Si hablamos de precisión, se ve en la Tabla 4.41, que se obtienen unos valores muy elevados para la categoría moreno, mientras que con las otras categorías la precisión es extremadamente baja. Esto también viene derivado de la imprecisión de las coordenadas, ya que si la zona de pelo coge algo más que no sea pelo, modificará los valores reales de pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

La evaluación del algoritmo de etiquetado con la cara recuadrada manualmente (véase Tabla 4.27), daba uno de los mejores resultados obtenidos en todo el proyecto, con lo que se ve la importancia de tener las coordenadas de la manera más precisa.

4.3.1.4 Barba

Ahora se pasa a etiquetar la presencia o ausencia de barba de las caras detectadas de la base de datos A. Se trata de una decisión binaria, con dos etiquetas: con barba y sin barba.

Esta es una característica muy importante para el posterior etiquetado de sexo y su matriz de confusión se muestra en la Tabla 4.42.

		VALOR PREDICHO	
		<u>BARBA</u>	<u>SIN BARBA</u>
VALOR REAL	<u>BARBA</u>	1	1
	<u>SIN BARBA</u>	9	36

Tabla 4.42: Matriz de confusión del etiquetado de barba.

En las 47 caras detectadas correctamente de la base de datos A, solo dos de ellas tienen barba. Como puede verse, en el caso de las personas sin barba, se etiquetan casi todas bien. Esto se puede comprobar en la Tabla 4.43.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
<i>Barba</i>	1	1	1	0.5
<i>Sin barba</i>	36	9	9	0.8

Tabla 4.43: Evaluación del etiquetado de barba.

La precisión para la categoría con barba es 0.5, debido a que etiqueta bien una de las dos posibles, mientras que para la categoría de sin barba, la precisión aumenta. Hay que tener en cuenta, que esto no es estadísticamente significativo, ya que solo existen dos imágenes, por lo que no se podrían extraer conclusiones.

Si comparamos estos valores con los de la Tabla 4.29, vemos que en este caso, no hay tanta diferencia entre los resultados obtenidos, lo que es muy útil debido a que esta etiqueta se necesita luego para el etiquetado de sexo.

4.3.1.5 Longitud de pelo

La siguiente característica es la longitud del pelo, que al igual que la etiqueta anterior se utilizará su resultado para el etiquetado de sexo.

Es también una decisión binaria, con dos etiquetas: largo y corto, que vienen representadas por la matriz de confusión de la Tabla 4.44.

		VALOR PREDICHO	
		<u>LARGO</u>	<u>CORTO</u>
VALOR REAL	<u>LARGO</u>	8	4
	<u>CORTO</u>	14	21

Tabla 4.44: Matriz de confusión del etiquetado de longitud de pelo.

Como se puede comprobar, se etiqueta bastante bien, pero en la categoría de pelo corto, se muestra un poco de confusión con la categoría de largo. Esto puede estar causado, porque al coger la zona para mirar el pelo largo, y el fondo (normalmente la ropa de la persona) es de color oscuro, podría entender que se trata de pelo. Para una mayor comprensión se tiene la Tabla 4.45.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Largo</i>	8	4	4	0.666
<i>Corto</i>	21	14	14	0.6

Tabla 4.45: Evaluación del etiquetado de longitud de pelo.

La precisión en ambas categorías, se podría considerar como aceptable, aunque ésta es más baja que cuando se analizó el algoritmo por separado (Tabla 4.31), lo que afectará negativamente a la evaluación de la etiqueta de sexo.

4.3.1.6 Sexo

Por último se tiene la etiqueta de sexo, que como ya se mencionó repetidas veces, depende directamente de las otras etiquetas analizadas anteriormente.

Como se ha ido comprobando, la precisión en casi todas las características es menor que en el caso de test, por lo que influirá en la evaluación de sexo, reduciendo también su precisión.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Esta etiqueta es una decisión binaria, con dos etiquetas, hombre y mujer, pero existe una etiqueta más, llamada "no detectado" que se da en aquellas caras cuyas respuestas de las otras etiquetas (barba, calvo, longitud de pelo y factor de sexo) sean incongruentes entre sí. De esta manera se intenta hacer al sistema más robusto frente a posibles errores.

		VALOR PREDICHO		
		<u>HOMBRE</u>	<u>MUJER</u>	<u>NO DETECTADO</u>
VALOR REAL	<u>HOMBRE</u>	17	10	2
	<u>MUJER</u>	9	7	2

Tabla 4.46: Matriz de confusión del etiquetado de sexo.

Como se ve en la Figura 4.46, en la categoría de hombre se etiquetan muchos aciertos, mientras que en la de mujer está más difusa la diferencia entre categorías.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
<i>Hombre</i>	17	12	12	0.586
<i>Mujer</i>	7	11	11	0.388

Tabla 4.47: Evaluación del etiquetado de sexo.

En la Tabla 4.47 se muestran los resultados de la evaluación de sexo y como puede comprobarse, la precisión no es muy elevada en ninguno de los casos. A diferencia de los resultados de analizar el algoritmo independientemente, Tabla 4.33, en este caso las categorías tienen una precisión parecida. Esto puede ser debido a que anteriormente, era más fácil detectar un hombre (si tiene barba ó esta calvo, por ejemplo) que a una mujer, pero ahora al estar afectado por los errores de las anteriores etiquetas, la fiabilidad con la que se etiquetaba a los hombres es menor.

4.3.2 DETECCIÓN BASADA EN PLANTILLAS Y ETIQUETADO

De igual manera que con el otro algoritmo, ahora se pasa a evaluar el sistema completo, pero con la detección basada en plantillas más el etiquetado. Se ejecutará este código para las 50 imágenes de la base de datos A, obteniendo la cara, si la hay, recuadrada y utilizando sus coordenadas para etiquetar dicha cara.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Como ya se ha evaluado el algoritmo de detección basado en plantillas con la base de datos A en el apartado 4.1, se sabe que de las 76 caras que hay repartidas a lo largo de toda la base de datos, este método detecta también 47 correctamente. Se analizará las etiquetas de manera independiente de todas las caras detectadas, para ver su precisión y si existe confusión entre las distintas categorías.

4.3.2.1 Raza

Se empieza evaluando la primera etiqueta, que es la raza y se tienen como categorías: blanco y negro.

Se crea la matriz de confusión, cuyos valores predichos se sitúan en el eje "x" y cuyas categorías están dispuesta en el eje "y", como se ve en la Tabla 4.48.

		VALOR PREDICHO	
		<u>BLANCO</u>	<u>NEGRO</u>
VALOR REAL	<u>BLANCO</u>	24	20
	<u>NEGRO</u>	1	2

Tabla 4.48: matriz de confusión del etiquetado de raza.

Como puede comprobarse en el caso de personas negras falla una de cada dos personas, mientras que para las personas blancas falla bastante más. Esto es debido a la imprecisión de las coordenadas, ya que si no son las correctas, no se cogería bien la zona de la frente por lo que los umbrales variarían.

En términos de precisión, en la Tabla 4.48, se puede ver esto mismo, ya que da algo mayor para el caso de personas negras, siendo bastante bajo para las personas blancas.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Blanco</i>	24	20	20	0.545
<i>Negro</i>	2	1	1	0.666

Tabla 4.49: Evaluación del etiquetado de raza.

Si se compara esta tabla con la obtenida en la Tabla 4.23 de la fase de etiquetado, se puede ver un gran decremento de las precisiones, por lo que se comprueba la importancia de las coordenadas correctas.

4.3.2.2 Color de ojos

La siguiente característica es el color de ojos, que se evaluará de igual manera que se explicó en 4.2.2.3.

Hay que tener en cuenta, que la forma de analizar si una etiqueta es cierta o no, se hace de manera subjetiva, dependiendo de la persona que evalúa el sistema.

		VALOR PREDICHO				
		<u>NEGROS</u>	<u>MARRONES</u>	<u>VERDES</u>	<u>AZULES</u>	<u>MAL ETIQUE.</u>
VALOR REAL	<u>NEGROS</u>	5	0	2	6	3
	<u>MARRONES</u>	0	10	2	7	2
	<u>VERDES</u>	0	0	5	0	0
	<u>AZULES</u>	0	0	0	0	1
	<u>Prob etiqueta</u>	0	0	0	2	2

Tabla 4.50: Matriz de confusión del etiquetado de color de ojos.

En la Tabla 4.50, se tiene la matriz de confusión donde se muestran los diferentes resultados del análisis. Como puede comprobarse, cuando se tienen ojos verdes se acierta siempre, pero existen bastantes errores tanto en ojos negros como en marrones, con las etiquetas verdes y azules. Esto es lo mismo que decir que existe cierta confusión con estas etiquetas.

El motivo de que sea con las etiquetas verde y azul, viene promovido por el hecho de que en este algoritmo para decidir por una etiqueta u otra, se hace la correlación por componentes R, G y B, y se mira cuándo se obtiene la suma más alta. Si yo correlaciono mi componente azul o verde, G o B, con la plantilla de dicho color, siempre me darán mayores resultados, por lo que el conjunto de la suma sería mayor y por tanto se etiquetará como si fuera esa la etiqueta correcta.

En referencia a la precisión, como indica la Tabla 4.51, se tiene la mayor precisión posible para los ojos verdes, pero en el resto de los casos se dan valores muy bajos.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
Negros	5	11	11	0.312
Marrones	10	11	11	0.476
Verdes	5	0	0	1
Azules	0	1	0	0
Prob. etiquetado	2	2	2	0.5

Tabla 4.51: Evaluación del etiquetado de color de ojos.

Si comparamos los resultados obtenidos para este algoritmo con la fase de etiquetado del color de ojos de la Tabla 4.25, se ve un empeoramiento claro. Esto se debe a que en la base de datos B que se usa para el entrenamiento del etiquetado, se tienen unas imágenes en ambiente controlado, donde la cara ocupa la mayor parte de la foto, no están giradas, solo existe una persona, etc. Esto hace que los ojos de las personas estén vistos de frente y con un tamaño aceptable. Para la base de datos A, al no tener un ambiente controlado, no se tienen por qué ver los ojos enteros, puede estar siendo captado desde otro ángulo de orientación, pueden existir varias personas en la misma foto, la cara no ocupa la mayor parte de la cara, etc. Esto provoca que los ojos no estén donde indican las coordenadas establecidas o que éstos sean tan pequeños que no se pueda obtener su color correctamente.

4.3.2.3 Color de pelo

Este algoritmo detecta el color de pelo de cada cara detectada y se tienen las etiquetas: moreno, castaño, rubio, pelirrojo y calvo.

Como se comentó en el apartado 4.3.1.3 en la detección de bajo nivel y etiquetado, existen personas canosas que no tienen una etiqueta como tal, por lo que se etiquetarán al color que más se asemejen.

		VALOR PREDICHO				
		<u>MORENO</u>	<u>CASTAÑO</u>	<u>RUBIO</u>	<u>PELIRROJO</u>	<u>CALVO</u>
VALOR REAL	<u>MORENO</u>	31	0	1	1	1
	<u>CASTAÑO</u>	2	2	0	0	2
	<u>RUBIO</u>	3	0	0	0	2
	<u>PELIRROJO</u>	1	0	0	0	0
	<u>CALVO</u>	0	1	0	0	0

Tabla 4.52: Matriz de confusión del etiquetado del color de pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Si se mira en la Tabla 4.52, se puede ver que se aciertan casi todas las personas de pelo moreno, es decir la mayoría de las fotos, y además existe poca confusión con las otras categorías.

Hay que resaltar el hecho de que se etiqueten varias personas de diferentes colores de pelo como calvas. Esto pasa por ejemplo con la imagen "36.jpg" (ver apartado 4.4.1), donde se tiene a dos hombres fotografiados desde arriba, por lo que al buscar la zona del pelo encima de las coordenadas de la cabeza y estar estas inclinadas, no encuentra el pelo, sino el fondo de la imagen. Otro ejemplo de etiquetado como calvo cuando no lo es, se da en la imagen "32.jpg", donde se tiene a una persona canosa, pero por las coordenadas que se dan poco precisas unidas al color cano que no detecta, entiende que no pertenece a ninguno de los umbrales de color.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Moreno</i>	31	3	3	0.911
<i>Castaño</i>	2	4	4	0.333
<i>Rubio</i>	0	5	5	0
<i>Pelirrojo</i>	0	1	1	0
<i>Calvo</i>	0	1	1	0

Tabla 4.53: Evaluación del etiquetado de color de pelo.

En la Tabla 4.53, se puede comprobar el elevado valor de la precisión para personas morenas, mientras que con las castañas se falla mucho más. Si comparamos estos resultados con los vistos en la Tabla 4.27, que fueron unos de los mejores de todo el análisis, se ve que existe un empeoramiento. Como se ha mencionado anteriormente, se debe a los fallos en las coordenadas y al ambiente no controlado de la base de datos A.

4.3.2.4 Barba

La siguiente etiqueta detecta la presencia o ausencia de barba y será utilizada en el etiquetado de sexo.

		VALOR PREDICHO	
		<u>BARBA</u>	<u>NO BARBA</u>
VALOR REAL	<u>BARBA</u>	2	3
	<u>NO BARBA</u>	6	36

Tabla 4.54: Matriz de confusión del etiquetado de barba.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Su matriz de confusión, vista en la Tabla 4.54, muestra que se detectan prácticamente todas las personas sin barba correctamente, mientras que si tenemos personas con barba falla más. Existe confusión entre las personas con barba y la etiqueta no barba.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Barba</i>	2	5	5	0.4
<i>Sin barba</i>	36	6	6	0.857

Tabla 4.55: Evaluación del etiquetado de barba.

Como se puede observar en la Tabla 4.55, y ya se adelantó anteriormente, la precisión para las personas sin barba, que son la mayoría de las imágenes, es muy elevada. En el caso de las personas con barba esta es bastante pequeña. Una vez más, si las coordenadas están mal estipuladas, harán que no se tome la zona de la barbilla correctamente y por lo tanto que se etiquete de manera errónea.

Si se compara estos resultados con los obtenidos en la Tabla 4.29, en la fase de etiquetado de barba, se puede ver que la precisión de las personas con barba es mayor, aunque en el caso de las personas sin barba se obtienen aquí mejores resultados.

4.3.2.5 Longitud del pelo

De igual manera que con las etiquetas anteriores, ahora se tiene otra decisión binaria con etiquetas: largo y corto. Esta característica será otra de las empleadas para el etiquetado de sexo.

Su matriz de confusión viene representada en la Tabla 4.56, y como puede comprobarse se etiquetan correctamente la mayor parte de las imágenes de personas con pelo corto. Existe confusión para personas de pelo largo ya que las etiqueta como corto, lo cual está promovido por la imprecisión de las coordenadas que hace que se tome una zona incorrecta para mirar si tiene el pelo largo. Además hay que tener también en cuenta la heterogeneidad de la base de datos A así como el ambiente no controlado en el que se encuentran, pudiendo estar las imágenes giradas, con diferentes fondos, etc.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

		VALOR PREDICHO	
		<u>CORTO</u>	<u>LARGO</u>
<u>VALOR REAL</u>	<u>LARGO</u>	4	5
	<u>CORTO</u>	29	9

Tabla 4.56: Matriz de confusión del etiquetado de longitud de pelo.

Para una mejor comprensión se recurre a la precisión, explicada en la Tabla 4.57. Como puede verse, en el caso de personas de pelo corto se tiene una precisión bastante buena, mientras que para las personas de pelo largo, solo se etiquetan correctamente algo más de un 55%.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Largo</i>	5	4	4	0.555
<i>Corto</i>	29	9	9	0.723

Tabla 4.57: Evaluación del etiquetado de longitud de pelo.

Al comparar esta tabla con la fase de entrenamiento del etiquetado de longitud de pelo vista en la Tabla 4.31, se ve que disminuyen ligeramente los valores de precisión, aunque no en gran medida.

4.3.2.6 Sexo

Por último se tiene la etiqueta más compleja. Ésta, como ya se mencionó a lo largo del proyecto, se crea a partir de algunas de las etiquetas ya vistas, como barba, calvo, longitud de pelo y factor de sexo, por lo que se irán sumando los errores de todas, que afectarán a la etiqueta final de sexo.

		VALOR PREDICHO		
		<u>HOMBRE</u>	<u>MUJER</u>	<u>NO DETEC</u>
<u>VALOR REAL</u>	<u>HOMBRE</u>	19	9	0
	<u>MUJER</u>	12	5	2

Tabla 4.58: Matriz de confusión del etiquetado de sexo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En la Tabla 4.58, se tiene la matriz de confusión donde puede comprobarse que los hombres se etiquetan bastante acertadamente, mientras que en las mujeres existe confusión con la etiqueta hombre. Esto viene derivado de que las propiedades que componen el etiquetado de sexo, son más relevantes para etiquetar hombre. Por ejemplo, la presencia de barba, o la calvicie, implica que indudablemente es un hombre, pero su ausencia no implica que sea mujer. Por este motivo, es más fiable la detección de hombres.

Como se ha ido viendo en los apartados anteriores, las etiquetas que se usan para obtener el etiquetado de sexo, han tenido menor precisión que en la fase de entrenamiento de etiquetado, es decir han fallado más, por lo que se puede adelantar, que la precisión del etiquetado de sexo tendrá también una menor precisión. Esto se puede comprobar en la Tabla 4.59.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Hombre</i>	19	9	9	0.678
<i>Mujer</i>	5	14	14	0.263

Tabla 4.59: Evaluación del etiquetado de sexo.

Si se compara esta tabla con la perteneciente al etiquetado de sexo de la fase de entrenamiento, Tabla 4.33, se ve que existe un empeoramiento, que como ya se adelantó se debe a la acumulación de los errores de las etiquetas que componen la etiqueta de sexo.

4.3.3 COMPARACIÓN ENTRE MÉTODOS CON ETIQUETADO

Una vez se tienen ya evaluadas todas y cada una de las etiquetas para los dos métodos de detección de caras, basado en bajo nivel y basado en plantillas, se muestra un resumen de todas ellas para compararlos de manera visual.

Lo primero es calcular los verdaderos positivos, TP, los falsos positivos, FP, y los falsos negativos, FN, para cada etiqueta independientemente de las categorías que haya. Con esto se obtiene una precisión por cada etiqueta, tanto de la detección de bajo nivel, Tabla 4.60, como para la detección basada en plantillas, Tabla 4.61.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Raza</i>	23	24	24	0.489
<i>Color ojos</i>	19	28	28	0.404
<i>Color Pelo</i>	35	12	12	0.744
<i>Barba</i>	37	10	10	0.787
<i>Longitud de pelo</i>	29	18	18	0.617
<i>Sexo</i>	24	23	23	0.51
Precisión media				0.5918

Tabla 4.60: Evaluación de cada etiqueta para la detección de bajo nivel.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>Raza</i>	26	21	21	0.553
<i>Color ojos</i>	22	25	25	0.468
<i>Color Pelo</i>	33	14	14	0.702
<i>Barba</i>	38	9	9	0.808
<i>Longitud de pelo</i>	34	13	13	0.723
<i>Sexo</i>	24	23	23	0.51
Precisión media				0.6273

Tabla 4.61: Evaluación de cada etiqueta para la detección basada en plantillas.

Si se comparan ambas tablas, se ve que siguen un patrón aproximadamente parecido. Los mejores resultados de precisión se dan en el etiquetado de barba y los siguientes valores son o en el color de pelo o en la longitud del pelo, en ambos casos con resultados bastante elevados. En torno a 0.5 de precisión se encuentran las etiquetas de raza y sexo. Por último se tiene la precisión del etiquetado de ojos, que en ambos casos es menor del 0.5.

También es útil comparar las Tablas 4.60 y 4.61, con la que se obtuvo en la fase de etiquetado para todas las etiquetas, Tabla 4.34. En general, en la fase de entrenamiento, todos los resultados son mayores de 0.6, siendo esta precisión la más pequeña también, en el etiquetado del color de ojos.

Para que se quede representado de manera visual se utiliza un diagrama de barras, Figura 4.5, donde se muestra la precisión de todas las etiquetas, estando en color azul el método de bajo nivel, en color rojo el método basado en plantillas y en color verde la fase de entrenamiento del etiquetado con la base de datos B.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

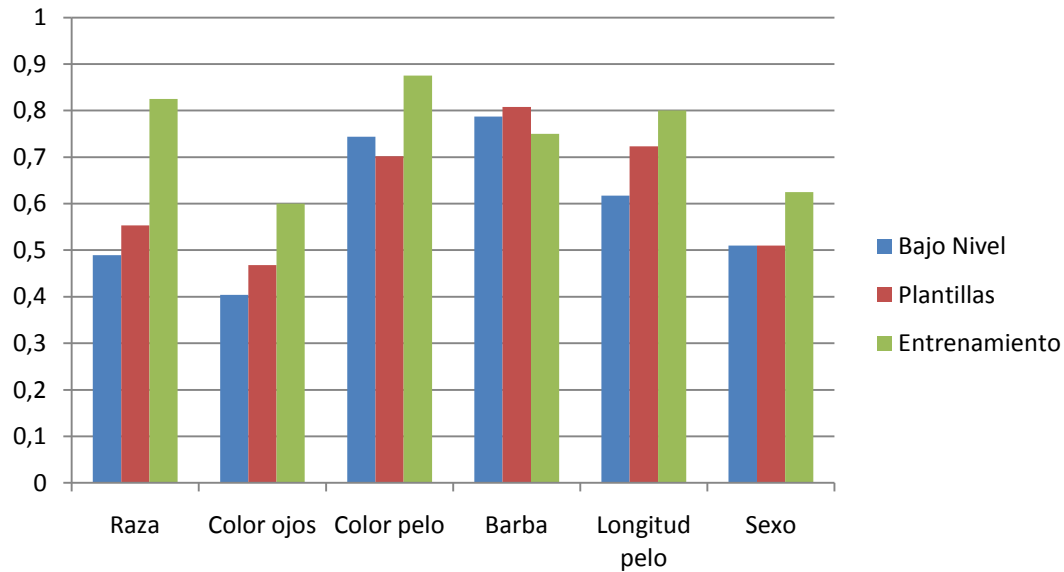


Figura 4.5: Diagrama de barras de la precisión de las etiquetas.

Como se puede observar en la Figura 4.5, para todas las etiquetas los mejores valores de precisión se dan en la fase de entrenamiento, como es obvio. Si solo nos fijamos en los métodos de detección, se puede comprobar, que el algoritmo basado en plantillas es el que mejor resultados da para todas las etiquetas, excepto para la del color de pelo.

Una forma sencilla para comparar las tres tablas y el diagrama de barras, es mediante el promedio de la precisión de todas las etiquetas, que se obtiene sumando la precisión de cada etiqueta y dividiendo este valor entre el total de etiquetas. Los resultados para la fase de etiquetado con la base de datos B, para la detección de bajo nivel con etiquetado y para la detección basada en plantillas con etiquetado, ambas con la base de datos A, se puede ver en la Tabla 4.62.

	<u>Precisión media</u>
<i>Entrenamiento etiquetado</i>	0.745
<i>Bajo nivel + etiquetado</i>	0.5918
<i>Plantillas + etiquetado</i>	0.6273

Tabla 4.62: Precisión media de todas las etiquetas para entrenamiento y test (detección de bajo nivel y basada en plantillas).

Como se puede comprobar en la tabla anterior, la mejor precisión media para todas las etiquetas, se da en la fase de etiquetado con la base de datos B, lo cual es obvio, ya que se entrenó con esa base de datos y además se le pasaban las coordenadas de la cara manualmente.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Entre los otros dos métodos de detección con el etiquetado, se obtiene una precisión media mayor en el caso de la detección basada en plantillas. Esto quiere decir, que habiendo detectado los dos métodos el mismo número de caras correctamente, 47, el de detección basado en plantillas ofrece mejores resultados a la hora de etiquetar esas caras, es decir, es más preciso en cuanto a seleccionar la región exacta de la cara (sus coordenadas).

Por último, se evalúa cada imagen en función del número de etiquetas que devuelve correctamente. Esto se hace basándonos en el anexo 6.4 y 6.5, donde de las 47 caras detectadas se señalan: sobre un fondo amarillo los resultados que tengan todas las etiquetas marcadas correctamente, sobre un fondo azul las imágenes que tengan 5 de las 6 etiquetas correctas, sobre un fondo verde las imágenes que tengan 4 etiquetas ciertas y sobre un fondo rosa las que solo tiene 3 etiquetas correctas.

Para el método de **detección de bajo nivel**, como se ve en la Tabla 4.63, se tiene una única cara detectada que devuelva las 6 etiquetas correctamente, 6 imágenes con 5 etiquetas correctas, 18 imágenes con 4 etiquetas correctas, etc.

Se puede comprobar que la mayor precisión se da para 4 etiquetas correctas, es decir, para este algoritmo se etiquetaran casi el 40% de las caras con 4 etiquetas correctas.

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>p</u>
6 etiquetas correctas	1	46	46	0.021
5 etiquetas correctas	6	41	41	0.127
4 etiquetas correctas	18	29	29	0.382
3 etiquetas correctas	11	36	36	0.234
0, 1 o 2 etiquetas correctas	11	36	36	0.234

Tabla 4.63: Número de imágenes con "x" etiquetas correctas para bajo nivel.

De igual manera, en la Tabla 4.64 se pueden ver los resultados de la **detección basada en plantillas**, donde se tienen 4 imágenes con todas las etiquetas correctas, 9 imágenes con 5 de las 6 imágenes correctas, 17 imágenes con 4 etiquetas correctas, etc.

De nuevo, la mayor precisión se da en las imágenes que tienen 4 etiquetas correctas de las 6 totales.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

	<u>TP</u>	<u>FN</u>	<u>FP</u>	<u>P</u>
<i>6 etiquetas correctas</i>	4	43	43	0.085
<i>5 etiquetas correctas</i>	9	38	38	0.191
<i>4 etiquetas correctas</i>	17	30	30	0.361
<i>3 etiquetas correctas</i>	9	38	38	0.191
<i>0, 1 o 2 etiquetas correctas</i>	8	39	39	0.17

Tabla 4.64: Número de imágenes con "x" etiquetas correctas para plantillas.

Al comparan estos valores con los obtenidos en la fase de entrenamiento de la base de datos B, en la Tabla 4.35, se ve que para ese caso, la mayor precisión se obtiene cuando se aciertan 5 etiquetas correctas.

Para saber cuando un sistema de detección y etiquetado es aceptable, se consideró que por lo menos la mitad de las etiquetas que se obtienen deberían ser correctas. Esto nos llevó a evaluar nuestro sistema en dos bloques, cuando se aciertan 4 o más de 4 etiquetas, o cuando se aciertan menos de 4 etiquetas. Sumando las imágenes que cumplen estas condiciones, según la Tabla 4.63 y 4.64, y sabiendo el número total de caras detectadas (para la base de datos A es 47 y para la base de datos B es 40), se obtuvieron los resultados representados en la Tabla 4.65.

	<u>Entrenamiento</u>	<u>Bajo nivel</u>	<u>Plantillas</u>
<i>4 o más etiquetas correctas</i>	32/40= 80%	25/47= 53.2%	30/47= 63.9%
<i>Menos de 4 etiquetas correctas</i>	8/40= 20%	22/47= 46.8%	17/47= 36.1%

Tabla 4.65: Evaluación en dos grupos en función del n° de etiquetas correctas.

Como puede comprobarse, para la fase de entrenamiento con la base de datos B, el 80% de las caras detectadas además aciertan como mínimo cuatro etiquetas. Cuando hablamos de los métodos de detección, sin tener las coordenadas manualmente, esos valores disminuyen, aunque a pesar de ello, en ambos algoritmos está por encima del 50%. En la detección de bajo nivel un 53.2% de las caras detectas se etiquetan con cuatro o más etiquetas correctas, mientras que en la detección basada en plantillas, este valor asciende hasta el 63.9%.

4.4 ANALISIS DEL ERROR

Hay una serie de fotos, que merecen una atención especial. Estas son, aquellas imágenes donde se obtienen resultados que no son los esperados, debido a características propias de las fotos o a errores que se arrastran y complican el funcionamiento de los algoritmos.

Se comentarán los resultados obtenidos en el conjunto de detección y etiquetado para la base de datos A, pero por comodidad se habla de ello en dos fases.

4.4.1 FASE DE DETECCIÓN

Mediante cualquier de los dos métodos de detección de caras, se obtienen las coordenadas y la cara enmarcada, si la hay, de cada imagen cuando cumplen ciertas condiciones, pero existen imágenes donde no se detecta la cara correctamente por sus características propias.

Como mencionamos en el apartado 2.1 no se consideran caras aquellas en las que se seleccione la cara junto con parte del pecho o fondo. Este es el caso de la foto 8 de la base de datos A.

Como puede verse en la figura 4.6, en la **detección de bajo nivel**, se detecta el pelo como píxeles de piel, por lo que luego tras la segmentación cumple todas las condiciones para validarla como cara. Esto se considera un falso positivo, ya que no está seleccionando solo la cara. En todas las pruebas que se han hecho a lo largo del proyecto, con cambios de umbrales, tamaño del elemento estructurante, etc, en ninguna de ellas se ha detectado correctamente.

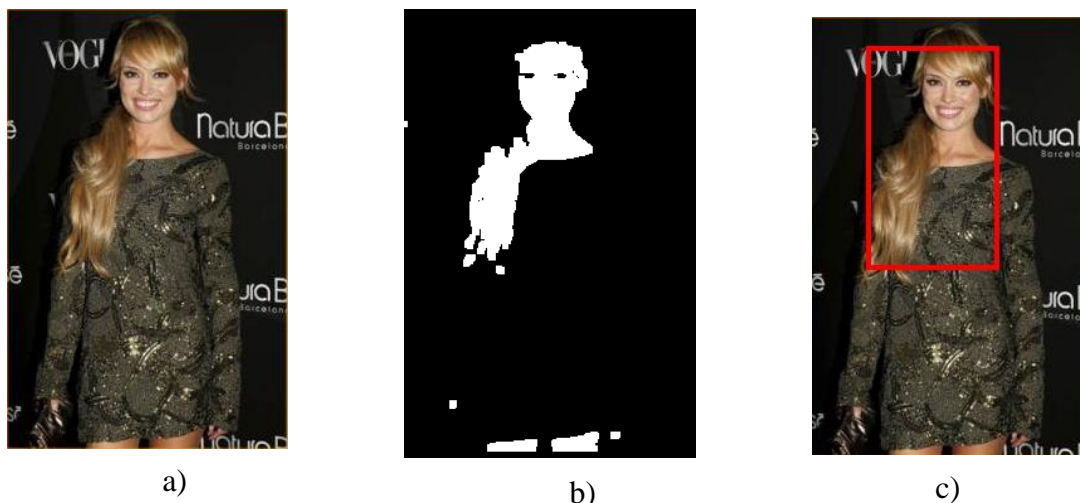


Figura 4.6: a) Imagen original; b) Imagen tras el filtrado; c) Imagen con la cara detectada;

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Con el **método basado en plantillas**, pasa algo parecido, ya que al detectar la piel, también detecta el pelo como píxel de piel. En este caso, al no cumplir la condición de correlación (se obtiene un valor de 0.552), no se marcaría como cara, como puede verse en la figura 4.7.

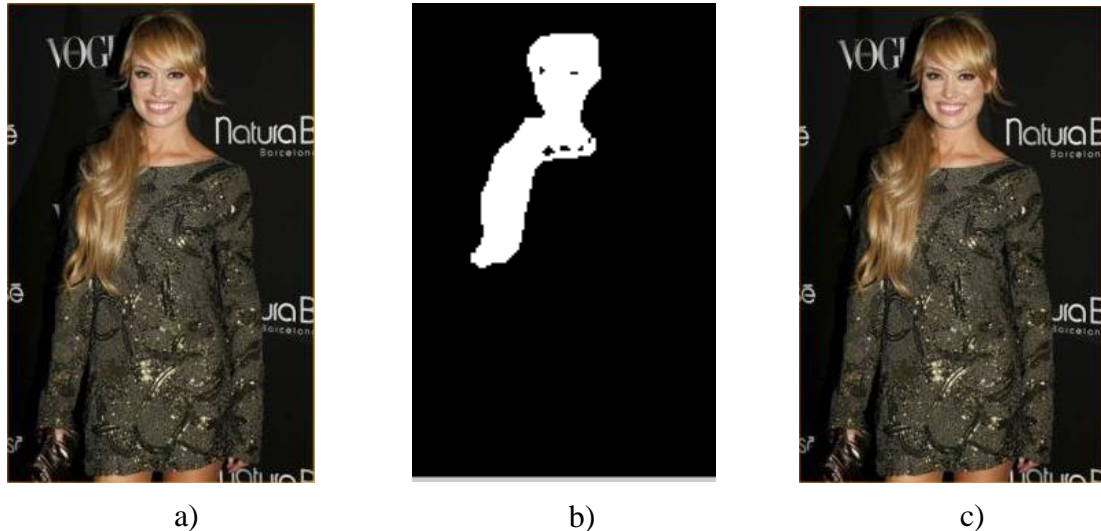


Figura 4.7: a) Imagen original; b) Imagen tras el filtrado;
c) Imagen con la cara detectada;

Otro caso parecido, es que no se consideran caras aquellas en las que se seleccione la cara junto al escote. Este es el caso de la foto 15, mostrada en la Figura 4.8, donde a pesar de haber probado con diferentes cambios, nunca se ha obtenido ninguno resultado satisfactorio.

En la **detección de bajo nivel**, como puede verse en la Figura 4.8, se seleccionan correctamente los píxeles de piel. Esto nos lleva al error, de segmentar la cara junto al escote, y al tener forma parecida a una elipse y cumplir el resto de condiciones, se etiquetará como cara.

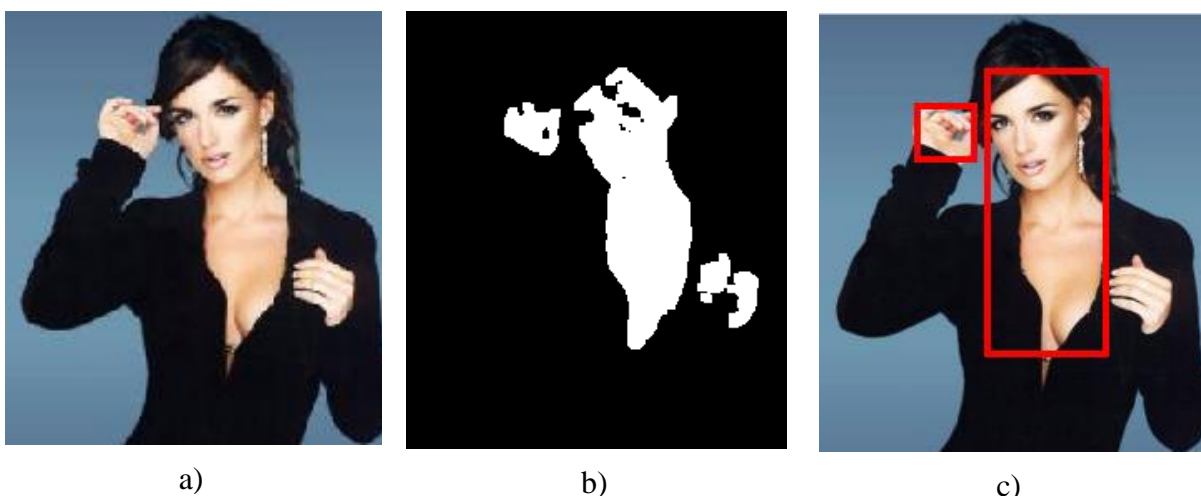


Figura 4.8: a) Imagen original; b) Imagen tras el filtrado;
c) Imagen con la cara detectada.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Con el algoritmo de **detección basado en plantillas** se detecta correctamente la piel de la imagen, es decir la cara y el cuello. En este caso, al no cumplir esta región la condición de correlación (se obtiene un valor de 0.5869), no se marca como cara.

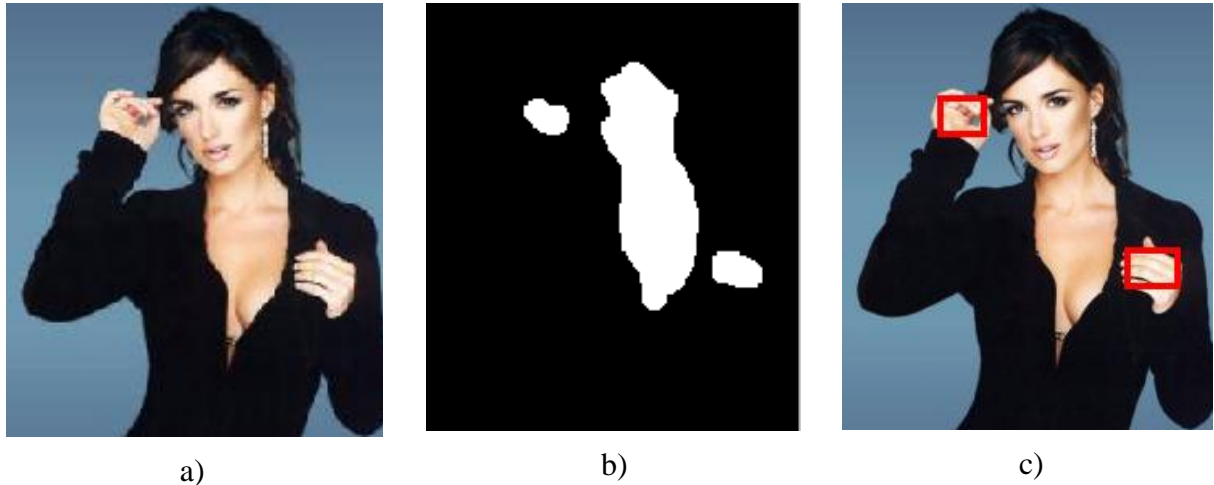


Figura 4.9: a) Imagen original; b) Imagen tras el filtrado; c) Imagen con la cara detectada.

De igual manera pasa con la foto 23 de la base de datos A, donde se tiene la cara y el pecho descubierto. En la **detección de caras de bajo nivel** a pesar de detectar correctamente la piel luego no cumple las condiciones para ser cara, como se ve en la Figura 4.10.

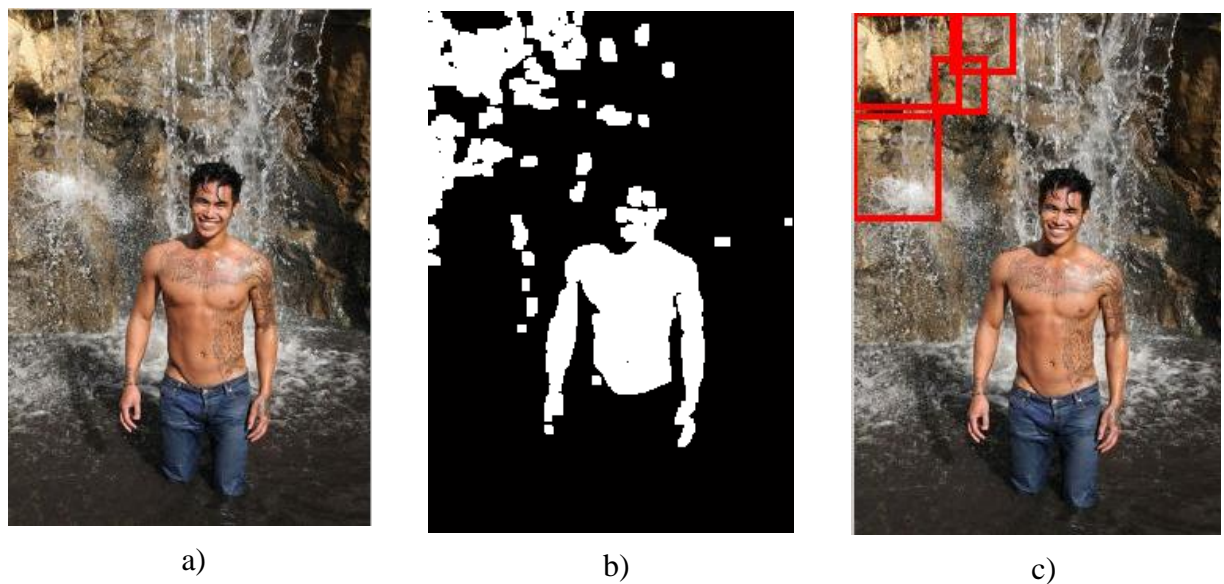


Figura 4.10: a) Imagen original; b) Imagen tras el filtrado; c) Imagen con la cara detectada.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En la **detección de rostros basada en plantillas** detecta la piel de forma correcta pero no cumple la condición de que su correlación sea mayor que 0.6, sino que obtiene un valor de 0.58.

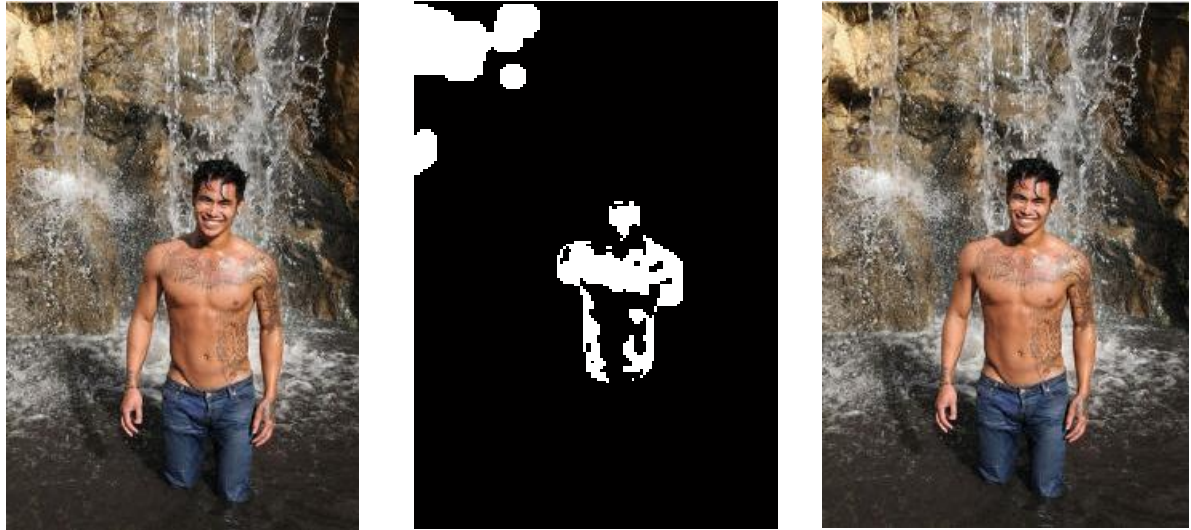


Figura 4.11: a) Imagen original; b) Imagen tras el filtrado; c) Imagen con la cara detectada.

Por último existe otro ejemplo donde aparte de tener cara y los hombros descubiertos, lleva unas gafas de sol que dificulta la tarea de detectar los rostros.

En el algoritmo de **detección de caras de bajo nivel** detecta correctamente la piel, poniendo a cero el espacio ocupado por las gafas. Independientemente de que se pudiera unir a la zona del pecho, la cara no cumple las condiciones necesarias de validación de rostros, ya que la región de la cara, está dividida por las gafas.

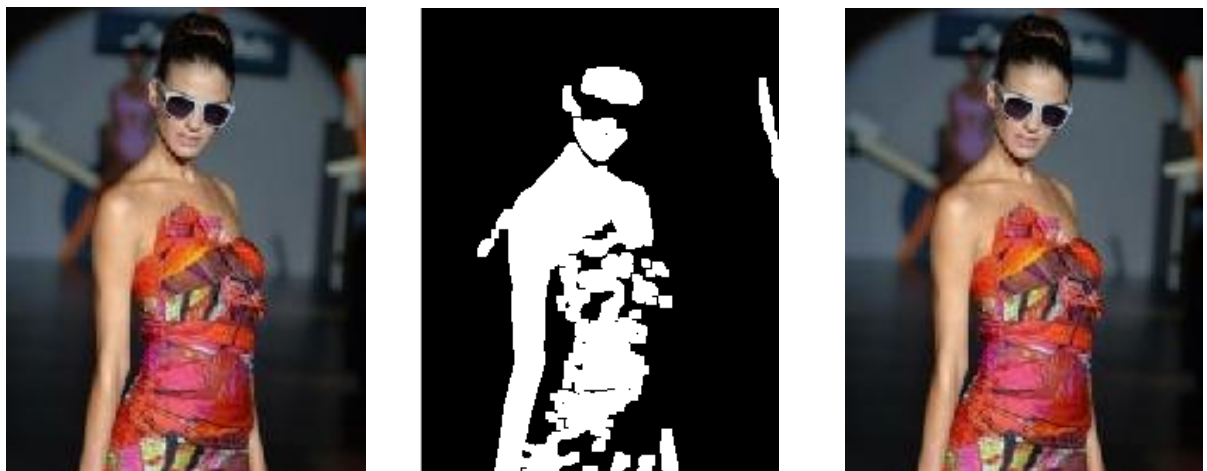


Figura 4.12: a) Imagen original; b) Imagen tras el filtrado; c) Imagen con la cara detectada.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En el caso de **detección basada en plantillas**, como se ve en la Figura 4.13, ocurre de igual manera que en el método anterior, ya que el espacio de las gafas divide a la zona del rostro en dos regiones, y ninguna de ellas cumple la condición de correlación (0.5751 y 0.493).



Figura 4.13: a) Imagen original; b) Imagen tras el filtrado; c) Imagen con la cara detectada.

Con todos estos ejemplos se ha comprobado, que cualquiera de los dos métodos, a pesar de detectar correctamente la piel y de que las condiciones de validación de caras sean bastante acertadas, hay casos en los que las características de la imagen suponen un impedimento en la detección. Por ejemplo, es cierto que una cara junto a un escote, se considera todo piel, lo cual es verdad, pero sin tener información externa no podríamos discriminar la región de piel de la cara, de la del pecho.

4.4.2 FASE DE ETIQUETADO

De igual manera, cuando se tiene la cara detectada y se etiqueta, hay veces que se producen errores y se tiene que saber porque vienen motivados.

El primero de ellos se da en el etiquetado de **raza**. Como ya se comentó, está basado en un umbral de la luminancia, para distinguir entre personas blancas y negras. Se escoge una zona de la frente, se calcula la media de la luminancia y se compara con unos umbrales. Pero hay veces que una misma persona, según la foto, puede verse con una tonalidad u otra, derivado de el uso de flash, del reflejo del sol, etc.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Esto pasa en varias imágenes, por ejemplo para la imagen "39.jpg" de la base de datos A en el método de bajo nivel, donde se tiene a una persona negra, pero en la frente el color de piel es más claro por lo que se etiqueta como blanco, siendo así un error.

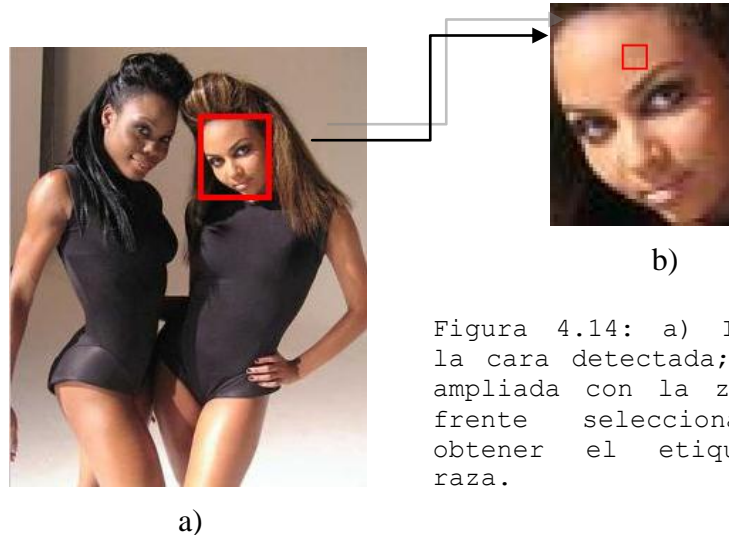


Figura 4.14: a) Imagen con la cara detectada; b) Cara ampliada con la zona de la frente seleccionada para obtener el etiquetado de raza.

Otro error típico, es que las coordenadas que devuelve el método de detección no sean tan precisas como debería, y así la zona de la frente, ya podría no ser tal, por lo que se etiquetaría incorrectamente. Eso pasa, por ejemplo en la imagen "34.jpg" para el método basado en bajo nivel, donde tenemos una persona blanca, pero selecciona mal la zona de la frente, por lo que la etiqueta como negro. Indudablemente esto se considerará un error.

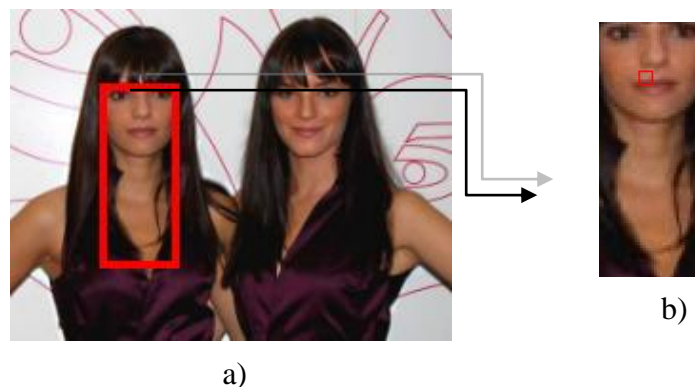


Figura 4.15: a) Imagen con la cara detectada; b) Cara ampliada con la zona de la frente seleccionada para obtener el etiquetado de raza.

El error por culpa de la mala precisión de las coordenadas influye en todos los umbrales que se tienen para obtener las etiquetas, por lo que se van a ir viendo ejemplos donde se da este caso.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

En el etiquetado del **color del pelo** como se mencionó en el apartado 4.3.2.3, es muy importante la precisión de las coordenadas así como que las imágenes no estén muy giradas. Un ejemplo de ello, sería la foto "36.jpg" donde se tienen a dos hombres y la fotografía está tomada desde arriba, por lo que al buscar encima de la cara la región de pelo, se encuentra con el fondo de la imagen. Como puede verse en la Figura 4.16, al ejecutar el método basado en plantillas, se tiene un hombre de pelo rubio que se etiqueta como moreno, debido a que coge parte del fondo como si fuera pelo.

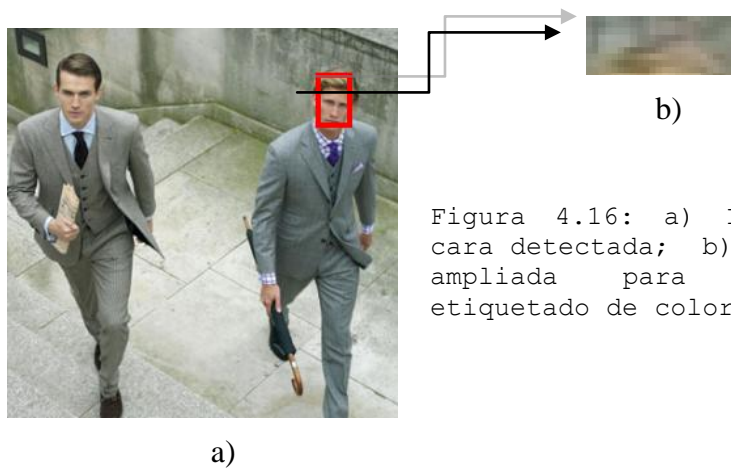


Figura 4.16: a) Imagen con la cara detectada; b) Zona del pelo ampliada para obtener el etiquetado de color de pelo.

Otro fallo derivado de las coordenadas para el color del pelo, se da en las personas calvas, en las cuales etiqueta como cara toda la zona, incluyendo donde estaría el pelo, por lo que al comprobar los umbrales para el color de pelo, cogería el fondo de la imagen, produciendo un error. Eso pasa en la fotografía "31.jpg", donde se tiene a una persona calva, y al coger la zona de pelo, selecciona el fondo, por lo que lo etiqueta incorrectamente.

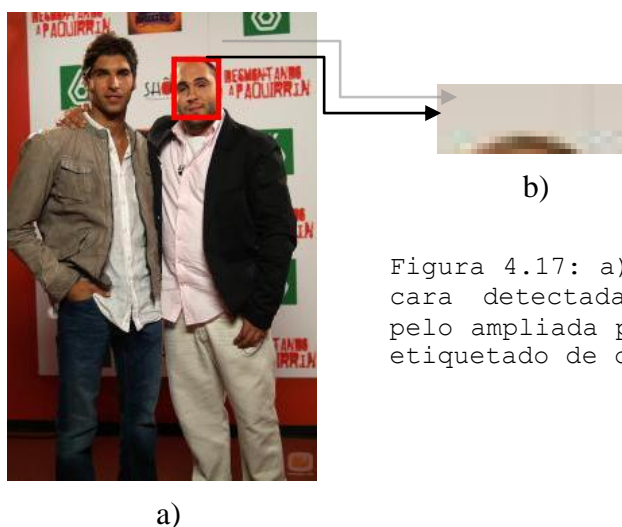


Figura 4.17: a) Imagen con la cara detectada; b) Zona de pelo ampliada para obtener el etiquetado de color de pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Otra etiqueta donde también es muy importante las coordenadas que se dan de la cara de la persona y si está girada o no, es la **barba**. En la Figura 4.18, la imagen "19.jpg" evaluada con el método basado en plantillas, se puede comprobar cómo al tener una cara girada, las coordenadas no son muy correctas, por lo que etiqueta como barba algo que no lo es.

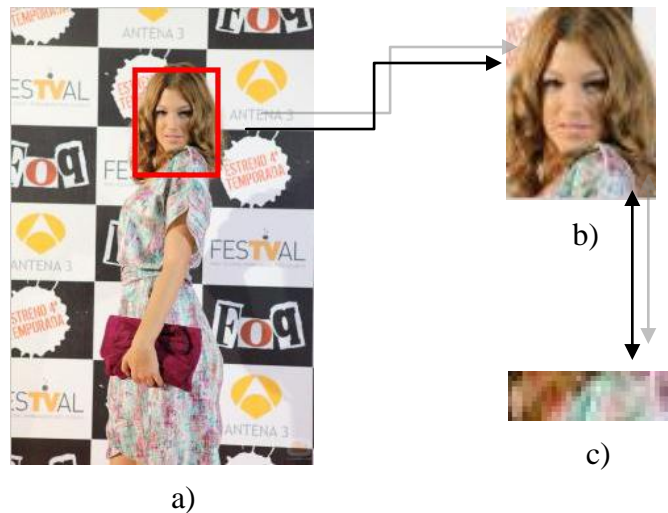


Figura 4.18: a) Imagen con la cara detectada; b) Cara detectada; c) Zona de la barbilla ampliada para obtener el etiquetado de la barba.

Por último en el etiquetado de la **longitud del pelo**, una mala precisión de las coordenadas, también puede crear errores de etiquetado. En la Figura 4.19, se tiene la imagen "30.jpg" y se evalúa con el método basado en plantillas. Hay una persona de pelo corto, pero al tener las coordenadas de la cara cogiendo parte del cuello, elige mal la región para comprobar si existe pelo largo, de manera que lo etiqueta como pelo largo, cuando esto es un error.

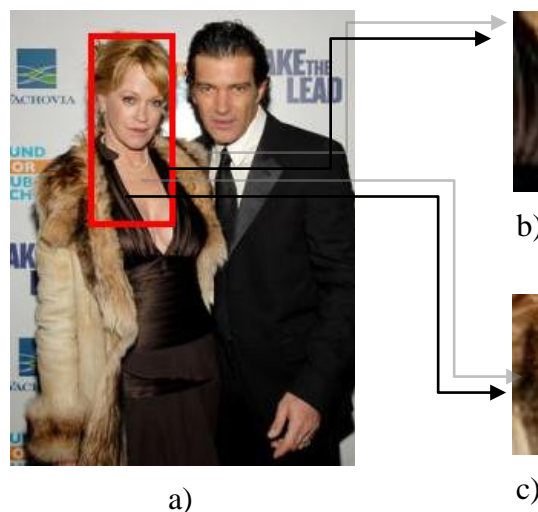


Figura 4.19: a) Imagen con la cara detectada; b) Recorte derecho ampliado para el etiquetado de longitud de pelo; c) Recorte izquierdo ampliado para el etiquetado de longitud de pelo.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

Con estos ejemplos se ha intentando demostrar, la importancia de una buena detección, tanto a la hora de seleccionar las caras como a la hora de escoger las coordenadas para describirla. Además la fase de etiquetado está condicionada a las imágenes que se emplean, por lo que aquí al usar la base de datos A y no estar en un ambiente controlado, dificultará su labor.

5. CONCLUSIONES Y TRABAJOS FUTUROS

5.1 CONCLUSIONES

El objetivo básico de este proyecto era el estudio de técnicas de detección y etiquetado de imágenes, la cuales están en un ambiente no controlado.

Se ha desarrollado un sistema en MATLAB, donde se tienen dos métodos para la detección de caras y otro método para el etiquetado de dichas caras.

El primer algoritmo de detección está basado en características de bajo nivel, donde se trabaja con el espacio de color RGB para obtener las regiones de piel y luego se aplican una serie de condiciones para decidir si esas regiones, pertenecen o no a una cara. Cuando detecta una cara, la selecciona y devuelve las coordenadas del recuadro que la contiene.

El otro método está basado en plantillas, de manera que se obtienen las regiones de piel en el espacio de color YCbCr, se aplican unas condiciones mínimas para que puedan considerarse cara y por último se correlaciona con una plantilla robotizada de una cara. Si detecta una cara, la selecciona y devuelve sus coordenadas.

El algoritmo de etiquetado, se ha probado primeramente con una base de datos B, en las cuales se daban las coordenadas de la cara manualmente. Con esta fase de entrenamiento, se ha obtenido una precisión media de 74.5% para las seis etiquetas: raza, color de ojos, color de pelo, barba, longitud de pelo y sexo.

Por último, a los dos métodos de detección se le añade la parte de etiquetado, donde se usan las coordenadas obtenidas en la detección de la cara, con las cuales se van extrayendo características para cada etiqueta. Dichas características se compararán con plantillas de las otras bases de datos, para así poder etiquetar a cada imagen en su categoría correspondiente. La salida que se obtiene con cada método, es la imagen con la cara, o caras recuadradas, junto con sus respectivas coordenadas y las seis etiquetas de cada una de ellas.

Si se compara entre los dos métodos de detección junto con el etiquetado para la base de datos A, los resultados que se obtienen son:

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

1. En la **fase de detección**, se obtiene el mismo número de caras detectadas con ambos algoritmos, un 61.8%, pero la cobertura obtenida por el método basado en plantillas es mayor, es decir, comete menos errores al etiquetar como caras regiones que no lo son. Esto implica que la media ponderada, F-measure, sea mayor también, con un 61.8%, comparada con la del método de bajo nivel que obtiene solo un 56.2%.

Si se habla de velocidad de respuesta, como se vio en la Tabla 4.10, el algoritmo de bajo nivel obtiene unos resultados medios muy favorables, de tan solo 1.396 segundos, mientras que con el segundo método, este tiempo aumenta hasta 7.957 segundos en el equipo de prueba.

Si se tienen limitaciones de tiempo de ejecución, a pesar de los mejores resultados obtenidos con la detección basada en plantillas, sería recomendable utilizar el otro algoritmo, ya que a igual precisión, disminuye la velocidad de respuesta media en una quinta parte.

2. En la **fase de etiquetado**, se tienen 47 caras detectadas correctamente tras el proceso de detección con cada uno de los métodos. La precisión media para el conjunto de las seis etiquetas, como se puede observar en la Tabla 4.61, es mayor en el algoritmo basado en plantillas, es decir, las coordenadas que devuelve este método son más precisas para la obtención de las etiquetas.

Otro dato importante, es que en la detección basada en plantillas, el 63.8% del total de las caras detectadas correctamente, aciertan como mínimo cuatro etiquetas de las seis posibles, mientras que en el algoritmo de bajo nivel, este valor se reduce a 53.1%.

En este proyecto se ha comprobado cómo el tipo de imágenes que se utilizan para la detección y el etiquetado es determinante. Cuantas más imágenes se tengan con condiciones adversas, como puede ser la aparición de múltiples personas en fotos de tamaño reducido o imágenes tomadas desde diferentes ángulos distintos al frontal, más complicado será su correcta detección.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

El etiquetado de las caras, está totalmente condicionado a las coordenadas obtenidas en la detección, como se puede ver si se comparan los resultados del etiquetado cuando las caras están detectadas manualmente y cuando se hace con las coordenadas que se obtienen de los métodos de detección. Este hecho está relacionado con un ambiente más o menos controlado en el que se adquieren esas fotografías.

De manera que si lo que se etiqueta como cara, en el sistema general de detección y etiquetado, coge solo parte de la parte cara o coge la cara unida al pelo, etc., las etiquetas se verían modificadas, pudiendo dar valores erróneos.

Por lo tanto, si se quiere implementar el sistema y obtener los mejores resultados posibles, se deberá tomar las imágenes en un ambiente relativamente controlado, en cuanto a personas giradas, luminosidad, número de personas por foto, etc., para que así las coordenadas de la cara sean lo más precisas y con ello se obtenga el mayor rendimiento posible con el etiquetado.

5.2 LINEAS FUTURAS DE TRABAJO

A la vista de los resultados han surgido varias líneas de continuación de las técnicas utilizadas en este proyecto, así como la creación de un único método de detección combinando los distintos algoritmos.

- Bases de datos de imágenes

Una posible mejora, buscar y seleccionar bases de datos "estándar" para comparación de algoritmos de clasificación y probar con ellas.

- Ajuste del tamaño de las imágenes

En las bases de datos seleccionadas se han empleado imágenes con tamaños relativamente pequeños para evitar una elevada velocidad de respuesta. Si el tiempo no es un requisito indispensable, se podría usar imágenes de una calidad superior con las cuales se detectarían mejor las caras en las fotografías que tienen un número elevado de personas.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- Preprocesado

En esta etapa se podrían utilizar más filtros para eliminar ruido así como añadir un método de compensación de iluminación.

- Validación de caras

El método utilizado para la detección de píxeles de piel en el algoritmo basado en plantillas, emplea el espacio de color YCbCr, que discrimina entre la luminancia y la crominancia, por lo que sería el método más preciso.

Tras esto, para la validación de caras, se podría crear un algoritmo combinando ambos métodos de detección y hacer un estudio para comprobar si existen otros parámetros que deban ser tenidos en cuenta a la hora de seleccionar las regiones de piel candidatas a ser caras.

- Mejora de las etiquetas

Para la etiqueta de la raza, se podrían seguir estudiando parámetros que permitieran la distinción entre otras categorías distintas a blanco y negro, como por ejemplo entre asiáticos.

Para el color de ojos, que es una de las etiquetas con peor precisión, se podrían buscar métodos de detección de iris y pupila, para que la correlación con las plantillas fuera más veraz.

Tanto para el etiquetado del color de pelo, de barba, y de longitud del pelo, se debería añadir algún parámetro que mejorara la búsqueda de la zona que se quiere etiquetar, de manera que si las coordenadas de la cara no son tan precisas como deberían, no tengan una gran influencia en la localización de dicha zona.

En la etiqueta del sexo, si se mejora cualquiera de los parámetros que la componen como se ha mencionado anteriormente, a la vez también aumentaría su precisión.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- Ampliación del número de etiquetas

En el proyecto se han obtenido seis etiquetas que sirven para describir a una persona, pero se pueden utilizar muchas otras que también podrían ser de una gran utilidad. Algunos ejemplos serían si una persona lleva bigote o gafas, si lleva gorros, si usa piercing o tatuajes, edad aproximada, peso y altura, etc.

- Análisis de expresiones fáciles

Otra posible mejora, sería añadir un sistema para analizar las expresiones faciales de la cara, que podrían servir también como etiquetas. No solo se especificaría si se tiene, por ejemplo, a un hombre blanco, con ojos azules, con el pelo largo y moreno, sino que se podría decir también si esa persona está alegre, enfadada o triste [9].

- Clasificación de las imágenes

Una vez se tiene ya la imagen con las caras detectadas y etiquetadas, se podría crear un sistema de almacenamiento de dichas imágenes, que se empleará como buscador de rostros en función de las etiquetas. De tal manera, que si se busca una mujer rubia y de ojos azules, apareciera un listado de todas las imágenes que contienen esa descripción.

6. ANEXOS

A continuación se muestran los valores obtenidos tras la ejecución de los diferentes algoritmos, tanto de detección como de etiquetado.

6.1. DETECCIÓN DE CARAS DE BAJO NIVEL

Resultados de la evaluación del método de bajo nivel con las 50 imágenes de la base de datos A.

<i>FOTO</i>	<i>Nº CARAS</i>	<i>TIEMPO</i>	<i>TP</i>	<i>FN</i>	<i>FP</i>
1	1	1.68	1	0	1
2	1	1.57	1	0	1
3	1	1.13	1	0	0
4	1	1.30	1	0	1
5	1	1.13	1	0	0
6	1	1.17	1	0	0
7	1	1.21	1	0	1
8	1	1.38	0	1	0
9	1	1.33	1	0	3
10	1	1.22	1	0	2
11	1	1.07	1	0	2
12	1	1.10	1	0	1
13	1	10.7	1	0	0
14	1	1.19	0	1	0
15	1	1.05	0	1	2
16	1	1.03	0	1	0
17	1	1.20	1	0	0
18	1	0.98	0	1	0
19	1	1.76	0	1	5
20	1	1.39	1	0	2
21	1	1.12	1	0	0
22	1	1.06	0	1	0
23	1	1.39	0	1	4
24	1	1.13	1	0	0
25	1	1.65	1	0	2
26	1	1.06	0	1	0
27	1	0.99	0	1	0
28	2	0.83	2	0	1
29	2	0.97	1	1	1
30	2	1.12	1	1	1
31	2	1.43	0	2	3
32	2	1.00	2	0	1
33	2	0.96	2	0	0
34	2	1.03	1	1	0

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

35	2	0.80	1	1	0
36	2	0.97	2	0	0
37	2	0.95	2	0	0
38	2	1.00	0	2	1
39	2	1.18	1	1	0
40	2	1.49	2	0	1
41	3	0.97	2	1	1
42	4	0.91	3	1	1
43	4	0.90	4	0	0
44	5	0.97	3	2	0
45	6	1.38	1	5	2
46	1	1.03	0	1	1
47	0	1.14	0	0	0
48	0	0.92	0	0	2
49	0	0.78	0	0	1
50	0	0.84	0	0	0
Suma de todas las imágenes			47	29	44

6.2 DETECCIÓN DE CARAS BASADA EN PLANTILLAS

Resultados de la evaluación del método basado en plantillas con las 50 imágenes de la base de datos A.

<i>FOTO</i>	<i>N° CARAS</i>	<i>TIEMPO</i>	<i>TP</i>	<i>FN</i>	<i>FP</i>
1	1	11.7	1	0	2
2	1	08.1	1	0	0
3	1	08.4	0	1	1
4	1	07.8	0	1	0
5	1	07.6	0	1	1
6	1	09.5	0	1	0
7	1	10.3	0	1	1
8	1	08.1	0	1	0
9	1	07.6	1	0	0
10	1	07.7	1	0	0
11	1	08.4	1	0	2
12	1	09.5	0	1	1
13	1	7.05	0	1	0
14	1	07.4	1	0	0
15	1	07.5	0	1	2
16	1	7.14	0	1	0
17	1	08.2	0	1	2
18	1	08.0	1	0	1
19	1	08.2	1	0	1
20	1	08.3	1	0	1
21	1	08.2	1	0	0

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

22	1	08.7	1	0	0
23	1	07.8	0	1	0
24	1	09.7	1	0	1
25	1	09.0	1	0	2
26	1	10.1	0	1	0
27	1	06.4	1	0	0
28	1	07.9	0	2	0
29	2	06.9	1	1	1
30	2	08.4	2	0	2
31	2	08.9	2	0	2
32	2	07.4	1	1	0
33	2	07.0	1	1	1
34	2	06.6	2	0	2
35	2	06.1	2	0	1
36	2	10.0	2	0	0
37	2	07.5	1	1	0
38	2	07.3	2	0	0
39	2	09.8	0	2	0
40	2	08.1	1	1	0
41	3	07.5	2	1	0
42	4	06.3	3	1	0
43	4	06.2	3	1	0
44	5	08.8	4	1	0
45	6	08.4	3	3	0
46	1	09.0	1	0	0
47	0	06.2	0	0	0
48	0	05.3	0	0	0
49	0	04.4	0	0	0
50	0	05.8	0	0	2
Suma de todas las imágenes			47	29	29

6.3 FASE DE ENTRENAMIENTO DEL ETIQUETADO

Resultados de las seis etiquetas obtenidas en la fase de entrenamiento del etiquetado con las 40 imágenes de la base de datos B. Indica las coordenadas que se le pasan manualmente, como $[x,y,w,h]$, siendo (x,y) las coordenadas del punto situado arriba y a la izquierda y (w,h) la anchura y la altura correspondiente a partir de ese punto.

El código de colores que se utiliza para clasificarlas es: si se tienen 6 etiquetas correctas, se subraya de amarillo; si se tiene 5 etiquetas correctas, se subraya de azul; si se tienen 4 etiquetas correctas, se subraya de verde; si se tienen 3 etiquetas correctas, se subraya de rosa y si se tienen menos de 3 etiquetas correctas, no se subraya.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

FOTO	COORDENADAS	ETIQUETAS
1	84,45,90,130	Blanco, ojos azules, moreno, barba, pelo largo, probabilidad hombre
2	41,110,163,195	Blanco, ojos verdes, moreno, sin barba, pelo corto, hombre
3	22,30,94,146	Blanco, ojos verdes, moreno, sin barba, pelo corto, probabilidad mujer
4	95,80,122,176	Blanco, ojos marrones, moreno, sin barba, pelo corto, hombre
5	77,135,212,266	Blanco, ojos marrones, moreno, sin barba, pelo corto, probabilidad mujer
6	45,38,92,131	Blanco, ojos marrones, calvo, sin barba, pelo largo, sexo mal etiquetado
7	52,34,63,80	Blanco, ojos azules, moreno, sin barba, pelo corto, hombre
8	65,52,82,110	Blanco, ojos negros, castaño, sin barba, pelo largo, probabilidad mujer
9	108,67,117,162	Blanco, ojos negros, castaño, sin barba, pelo corto, hombre
10	72,38,72,92	Blanco, ojos marrones, moreno, sin barba, pelo corto, hombre
11	161,53,130,171	Negro, ojos marrones, castaño, no barba, pelo largo, probabilidad mujer
12	40,40,68,87	Blanco, ojos verdes, moreno, sin barba, pelo corto, hombre
13	85,56,146,224	Blanco, ojos mal etiquetados, moreno, sin barba, largo, probabilidad mujer
14	50,55,91,115	Blanco, ojos negros, moreno, barba, pelo largo, prob hombre
15	32,32,104,150	Blanco, ojos verdes, moreno, sin barba, pelo corto, hombre
16	100,90,180,249	Blanco, ojos azules, moreno, sin barba, pelo corto, prob mujer
17	27,67,133,175	Blanco, ojos azules, moreno, sin barba, pelo corto, hombre
18	110,27,96,133	Blanco, ojos azules, moreno, barba, pelo largo, sexo mal etiquetado
19	41,21,52,63	Blanco, ojos verdes, pelirrojo, sin barba, pelo largo, probabilidad mujer
20	37,47,92,135	Blanco, ojos azules, moreno, sin barba, pelo largo, probabilidad mujer
21	62,34,89,137	Blanco, ojos marrones, moreno, barba, pelo corto, hombre
22	32,75,108,122	Negro, ojos problema etiquetado, moreno, sin barba, pelo corto, hombre
23	89,96,120,180	Negro, ojos negros, moreno, sin barba, pelo largo, probabilidad mujer
24	60,58,93,122	Blanco, ojos marrones, castaño, barba, pelo corto, hombre
25	80,66,98,132	Blanco, ojos azules, castaño, sin barba, pelo corto, hombre
26	80,68,120,150	Blanco, ojos marrones, moreno, sin barba, pelo largo, probabilidad mujer

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

27	40,55,45,54	Blanco, ojos negros, rubio, sin barba, pelo largo, probabilidad mujer
28	32,25,90,136	Blanco, ojos mal etiquetados, pelirrojo, barba, pelo corto, probabilidad hombre
29	76,34,76,108	Blanco, ojos mal etiquetados, castaño, barba, pelo largo, probabilidad hombre
30	54,25,90,110	Negro, ojos verdes, pelirrojo, sin barba, pelo corto, hombre
31	64,28,55,76	Blanco, ojos verdes, castaño, barba, pelo corto, probabilidad hombre
32	23,36,64,85	Blanco, ojos azules, calvo, barba, pelo corto, hombre
33	34,38,120,170	Negro, ojos azules, moreno, sin barba, pelo corto, hombre
34	25,26,80,110	Blanco, ojos negros, moreno, barba, pelo corto, hombre
35	40,15,82,117	Negro, ojos mal etiquetados, moreno, sin barba, corto, probabilidad mujer
36	24,23,54,70	Negro, ojos negros, calvo, sin barba, pelo corto, hombre
37	16,28,39,54	Negro, ojos mal etiquetado, moreno, sin barba, pelo corto, hombre
38	10,30,55,56	Negro, ojos azules, calvo, sin barba, pelo corto, hombre
39	95,45,90,135	Blanco, ojos marrones, moreno, barba, pelo largo, probabilidad hombre
40	25,38,76,105	Blanco, ojos azules, moreno, sin barba, pelo corto, hombre

6.4 DETECCIÓN DE BAJO NIVEL CON ETIQUETADO

Resultados de las seis etiquetas obtenidas de aplicar el método de bajo nivel y el etiquetado a las caras detectadas correctamente de la base de datos A. Cuando en una imagen hay varias caras detectas, se enumeran de izquierda a derecha.

El código de colores que se utiliza para clasificarlas es: si se tienen 6 etiquetas correctas, se subraya de amarillo; si se tiene 5 etiquetas correctas, se subraya de azul; si se tienen 4 etiquetas correctas, se subraya de verde; si se tienen 3 etiquetas correctas, se subraya de rosa y si se tienen menos de 3 etiquetas correctas, no se subraya.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

<u>FOTO</u>	<u>N° CARAS DETECTADAS BIEN</u>	<u>ETIQUETAS</u>
1	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
2	1	Negro, ojos marrones, moreno, sin barba, pelo corto, hombre
3	1	Blanco, ojos azules, moreno, barba, pelo corto, hombre
4	1	Blanco, ojos azules, moreno, barba, pelo corto, hombre
5	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
6	1	Negro, ojos verdes, pelirrojo, sin barba, pelo corto, hombre
7	1	Negro, ojos azules, moreno, sin barba, pelo corto, probabilidad mujer
8	0	No se detecta la cara
9	1	Negro, ojos verdes, moreno, sin barba, pelo corto, probabilidad mujer
10	1	Negro, ojos azules, calvo, sin barba, pelo largo, sexo mal etiquetado
11	1	Negro, ojos azules, moreno, sin barba, pelo corto, hombre
12	1	Blanco, ojos azules, moreno, barba, pelo largo, sexo mal etiquetado
13	1	Blanco, ojos verdes, moreno, barba, pelo largo, probabilidad hombre
14	0	No se detecta la cara
15	0	No se detecta la cara
16	0	No se detecta la cara
17	1	Negro, ojos azules, moreno, sin barba, pelo largo, mujer
18	0	No se detecta la cara
19	0	No se detecta la cara
20	1	Blanco, ojos azules, moreno, sin barba, pelo corto, hombre
21	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
22	0	No se detecta la cara
23	0	No se detecta la cara
24	1	Blanco, azules, moreno, barba, pelo corto, hombre
25	1	Negro, ojos azules, moreno, sin barba, pelo corto, hombre
26	0	No se detecta la cara
27	0	No se detecta la cara
28	2	1. Blanco, ojos verdes, pelirrojo, barba, pelo largo, probabilidad hombre 2. Blanco, ojos verdes, moreno, sin barba, pelo largo, probabilidad mujer
29	1	Negro, ojos mal etiquetados, moreno, sin barba, pelo largo, mujer

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

30	1	Blanco, ojos azules, moreno, sin barba, pelo corto, probabilidad mujer
31	0	No se detecta la cara
32	2	1. Blanco, ojos mal etiquetados, moreno, barba, pelo corto, hombre 2. Negro, ojos azules, moreno, sin barba, pelo largo, probabilidad mujer
33	2	1. Blanco, ojos mal etiquetados, moreno, barba, pelo corto, hombre 2. Blanco, ojos marrones, moreno, barba, pelo corto, hombre
34	1	Negro, ojos marrones, moreno, sin barba, pelo largo, probabilidad mujer
35	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
36	2	1. Blanco, ojos marrones, calvo, sin barba, pelo corto, hombre 2. Negro, ojos negros, calvo, sin barba, pelo corto, hombre
37	2	1. Blanco, ojos azules, moreno, sin barba, pelo largo, probabilidad mujer 2. Blanco, ojos azules, moreno, sin barba, pelo largo, probabilidad mujer
38	0	No se detecta la cara
39	1	Blanco, ojos azules, castaño, sin barba, pelo corto, hombre
40	2	1. Negro, ojos negros, calvo, sin barba, pelo corto, hombre 2. Negro, ojos marrones, moreno, sin barba, pelo largo, probabilidad mujer
41	2	1. Negro, ojos negros, calvo, sin barba, pelo corto, hombre 2. Negro, ojos verdes, moreno, sin barba, pelo largo, probabilidad mujer
42	3	1. Blanco, ojos azules, moreno, sin barba, pelo corto, hombre 2. Blanco, ojos verdes, moreno, sin barba, pelo corto, hombre 3. Negro, ojos verdes, moreno, sin barba, pelo largo, probabilidad mujer
43	4	1. Negro, ojos marrones, calvo, sin barba, pelo largo, sexo mal etiquetado 2. Blanco, ojos marrones, moreno, sin barba, pelo largo, probabilidad mujer 3. Negro, ojos negros, moreno, sin barba, pelo largo, probabilidad mujer 4. Negro, ojos negros, moreno, sin barba, pelo corto, hombre
44	3	1. Blanco, ojos azules, moreno, sin barba, pelo largo, probabilidad mujer 2. Blanco, ojos verdes, moreno, barba, pelo largo, probabilidad hombre 3. Blanco, ojos azules, calvo, sin barba, pelo largo, sexo mal etiquetado
45	1	Blanco, ojos verdes, moreno, barba, pelo corto, hombre
46	0	No se detecta la cara

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

47	0	No hay cara que detectar
48	0	No hay cara que detectar
49	0	No hay cara que detectar
50	0	No hay cara que detectar

6.5 DETECCIÓN BASADA EN PLANTILLAS CON ETIQUETADO

Resultados de las seis etiquetas obtenidas de aplicar el método basado en plantillas y el etiquetado a las caras detectadas correctamente de la base de datos A. Cuando en una imagen hay varias caras detectadas, se enumeran de izquierda a derecha.

El código de colores que se utiliza para clasificarlas es: si se tienen 6 etiquetas correctas, se subraya de amarillo; si se tiene 5 etiquetas correctas, se subraya de azul; si se tienen 4 etiquetas correctas, se subraya de verde; si se tienen 3 etiquetas correctas, se subraya de rosa y si se tienen menos de 3 etiquetas correctas, no se subraya.

<u>FOTO</u>	<u>N° CARAS DETECTADAS BIEN</u>	<u>ETIQUETAS</u>
1	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
2	1	Negro, ojos marrones, moreno, sin barba, pelo corto, hombre
3	0	No se detecta la cara
4	0	No se detecta la cara
5	0	No se detecta la cara
6	0	No se detecta la cara
7	0	No se detecta la cara
8	0	No se detecta la cara
9	1	Negro, ojos negros, moreno, sin barba, pelo corto, probabilidad mujer
10	1	Blanco, ojos azules, moreno, sin barba, pelo largo, mujer
11	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
12	0	No se detecta la cara
13	0	No se detecta la cara
14	1	Negro, ojos verdes, castaño, sin barba, pelo corto, hombre
15	0	No se detecta la cara
16	0	No se detecta la cara

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

17	0	No se detecta la cara
18	1	Blanco, ojos azules, moreno, barba, pelo largo, probabilidad hombre
19	1	Blanco, ojos azules, moreno, barba, pelo corto, hombre
20	1	Blanco, ojos azules, moreno, sin barba, pelo corto, hombre
21	1	Blanco, ojos marrones, moreno, barba, pelo corto, hombre
22	1	Blanco, ojos marrones, moreno, barba, pelo corto, hombre
23	0	No se detecta la cara
24	1	Negro, ojos azules, moreno, sin barba, pelo corto, hombre
25	1	Blanco, ojos mal etiquetados, moreno, barba, pelo corto, hombre
26	0	No se detecta la cara
27	1	Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
28	0	No se detecta la cara
29	1	Negro, ojos mal etiquetados, rubio, sin barba, corto, probabilidad mujer
30	2	1.Negro, ojos marrones, moreno, sin barba, pelo corto, probabilidad mujer 2.Blanco, ojos negros, moreno, barba, pelo largo, sexo mal etiquetado
31	2	1.Negro, ojos mal etiquetados, pelirrojo, sin barba, largo, mujer 2.Negro, ojos marrones, castaño, sin barba, pelo largo, mujer
32	1	Blanco, ojos mal etiquetados, calvo, sin barba, pelo corto, hombre
33	1	Blanco, ojos mal etiquetados, moreno, sin barba, pelo corto, hombre
34	2	1.Negro, mal etiquetados, moreno, sin barba, largo, probabilidad mujer 2.Negro, ojos marrones, moreno, sin barba, pelo largo, probabilidad mujer
35	2	1.Negro, ojos verdes, moreno, sin barba, pelo corto, hombre 2.Blanco, ojos verdes, moreno, barba, pelo corto, hombre
36	2	1.Blanco, ojos negros, calvo, sin barba, pelo corto, hombre 2.Negro, ojos azules, moreno, sin barba, pelo corto, hombre
37	1	Blanco, ojos azules, moreno, barba, pelo largo, probabilidad hombre
38	2	1.Blanco, ojos azules, moreno, sin barba, pelo corto, hombre 2.Blanco, ojos verdes, moreno, sin barba, pelo corto, hombre
39	0	No se detecta la cara
40	1	Negro, ojos mal etiquetados, calvo, sin barba, pelo largo, mal etiquetado

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

41	2	1.Negro, ojos azules, moreno, sin barba, pelo corto, hombre 2.Negro, ojos marrones, moreno, sin barba, pelo largo, probabilidad mujer
42	3	1.Blanco, ojos marrones, moreno, sin barba, pelo corto, hombre 2.Blanco, ojos negro, moreno, sin barba, pelo corto, hombre 3.Negro, ojos azules, moreno, sin barba, pelo corto, hombre
43	3	1.Blanco, mal etiquetados, moreno, sin barba, largo, probabilidad mujer 2.Blanco, ojos marrones, moreno, sin barba, pelo corto, hombre 3.Negro, ojos verdes, moreno, sin barba, pelo corto, hombre
44	4	1.Blanco, ojos azules, moreno, sin barba, pelo corto, hombre 2.Blanco, ojos azules, moreno, sin barba, pelo largo, probabilidad mujer 3.Blanco, ojos verdes, moreno, sin barba, pelo largo, probabilidad mujer 4.Blanco, ojos verdes, moreno, sin barba, pelo largo, probabilidad mujer
45	3	1.Negro, ojos azules, calvo, sin barba, pelo corto, hombre 2.Blanco, ojos mal etiquetados, calvo, sin barba, pelo corto, hombre 3.Blanco, ojos marrones, barba, pelo corto, hombre
46	1	Negro, ojos verdes, castaño, sin barba, pelo corto, probabilidad mujer
47	0	No hay cara que detectar
48	0	No hay cara que detectar
49	0	No hay cara que detectar
50	0	No hay cara que detectar

6.5 MANUAL DEL USUARIO

Para la utilización del sistema implementado se necesita tener la imagen o imágenes que se quieren etiquetar y disponer de las bases de datos para entrenar el sistema. Estas son bases de datos de: tipos de pieles, de tipos de pelo y de plantillas de ojos.

Para utilizar cualquier de los dos métodos que detectan las caras se debe realizar una llamada a la función **detectorCaraBajoNivel** o **detectorCaraPlantillas**, pasándole como parámetro la ruta completa donde se encuentra la imagen a detectar. Si detecta alguna cara, llama directamente a la función **etiquetado**. La salida que devuelven ambos métodos es el número de caras detectadas, las coordenadas de la posición de las caras en la imagen y las seis etiquetas pertenecientes a dichas caras.

6.6 CÓDIGO MATLAB

A continuación se presentan las funciones que se implementan para este proyecto. Están divididas en los tres algoritmos fundamentales, dentro de los cuales existen otras funciones básicas.

- Función ***detectorCaraBajoNivel***

```
function [numCaras, resultados, etiquetas]= detectorCaraBajoNivel(imagen)

img=imread(imagen);
imshow(img);

I=double(img);
H=size(I,1);
W=size(I,2);

R=I(:,:,1);
G=I(:,:,2);
B=I(:,:,3);

%%%%%%%% Filtro de mediana %%%%%%%%%%

filtrada(:,:,1) =medfilt2(R);
filtrada(:,:,2) =medfilt2(G);
filtrada(:,:,3) =medfilt2(B);

%%%%%%%%

%%%%%%%% Segmentación piel RGB %%%%%%%%%

maxi=max(filtrada,[],3); % Maximo en los 3 colores
mini=min(filtrada,[],3); % Mínimo en los 3 colores

% Se obtiene a partir de una imagen color-RGB una imagen binaria %donde los valores 1
corresponden a la presencia de píxeles de %piel.

% Condiciones
c1=(R>95) & (G>40) & (B>20) & ((maxi-mini)>15);
c2=abs(R-G)>15;
c3=(R>G) & (R>B);

% Detección píxeles de piel
piel=(c1&c2&c3);

figure();
imshow(piel)
title('Segmentacion piel');

%%%%%%%%

%%%%%%%% Erosion con un disco de radio 3 %%%%%%%%%

ee=strel('disk',3);
erosionada=imerode(piel, ee);

imshow(erosionada);
title('Imagen erosionada');

%%%%%%%%

%%%%%%%% Dilatacion con un disco de ratio 3 %%%%%%%%%

eel=strel('disk',3);
dilatada=zeros(H,W);
dilatada=imdilate(erosionada, eel);

%%Multiplica la imagen dilatada con la imagen segmentada
%%de piel para mantener los agujeros y luego pinta dicha imagen.

dilatada=immultiply(dilatada,piel);

imshow(dilatada);
title('Imagen dilatada');

% Etiquetado de las regiones
[imagen_etiquetada,num]=bwlabel(dilatada);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Validación de caras %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

propiedades=regionprops(imagen_etiquetada,'BoundingBox','MinorAxisLength','MajorAxisLength',
'Area','EulerNumber','Solidity','Image');

numCaras=0;
detectada=[];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Condiciones para ser cara %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i = 1:num
region=(imagen_etiquetada==i); %Selección de una región

%Calculo del perimetro, relación de aspecto, factor de forma y n° Euler
perimetro= bwperim(region);
area=bwarea(perimetro);
relacion_aspecto(i)=propiedades(i).MinorAxisLength/propiedades(i).MajorAxisLength;
factor_forma(i)=(4*pi)*(propiedades(i).Area/area^2);
euler=propiedades(i).EulerNumber;
agujeros=1-euler;

%%% Condiciones
condicion=(factor_forma(i)>0.1)&(relacion_aspecto(i)>0.3)&(propiedades(i).Solidity>0.7)&(agujeros>1);

if condicion
numCaras=numCaras+1;
detectada(numCaras).image=propiedades(i).Image;
detectada(numCaras).posicion=propiedades(i).BoundingBox;
end

end

numCaras
figure();
imshow(img);

%%% Matrices para guardar resultados
resultados=zeros(length(detectada),4);
etiquetas=zeros(length(detectada),6);

%%% Se obtienen las coordenadas de las regiones
for i=1:length(detectada)
coordenadas = round((detectada(i).posicion));
A=detectada(i).image;

% figure();
% imshow(A);
% title('Image');

x=coordenadas(1);
y=coordenadas(2);
w=coordenadas(3);
h=coordenadas(4);

%%% Se pinta un recuadro con las coordenadas de la cara
hold on
rectangle('Position',[x,y,w,h],'LineWidth',3, 'EdgeColor','r');
hold off

%%% Se guardan las coordenadas de cada región
resultados(i,:)=[x,y,w,h];

%%% Se llama a la función para el etiquetado
[colorPelo, colorOjos, calvo, barba, mujer, largo]=etiquetado(imagen, x, y, w, h);

%%% Se guardan los resultados del etiquetado
etiquetas(i,:)=[colorPelo, colorOjos, calvo, barba, mujer, largo];
end
```

- Función ***detectorCaraPlantillas***

```
function [NumCaras, resultados, etiquetas]=detectorCaraPlantillas(imagen)

close all;
img=imread(imagen);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
figure()
imshow(img);

% Crea un modelo de piel mediante imagenes de pieles.
[crmean,cbmean,crcbcov]=modelo_piel();
%Convierte la imagen de entrada en una imagen de probabilidad de %piel.
[probabilidad_piel]=probabilidad(imagen,crmean,cbmean,crcbcov);
%Segmentacion de la piel mediante umbral adaptivo
[piei_segmentada, umbral_optimo] = segmentacion(probabilidad_piel);
%Aplica tecnicas morfologicas y etiqueta regiones.
[regiones, num]=procesado_regiones(piei_segmentada);
%Aplica el test de euler.
[regiones_euler, num]=euler(regiones, num);
%Aplica el test de relacion.
[regiones_ratio, num]=ratio(regiones_euler, num);
%Aplica el test de plantilla.
[regiones_plantilla, RectCoord, num]=plantilla(regiones_ratio, imagen, 'plantilla.jpg',
num);

%Se coge la imagen en binario de las regiones y la multiplicamos %por la imagen(las tres
matrices) para obtener solo las caras en %color y el resto en negro.

for i=1:3
    imagenF(:, :, i)=(double(img(:, :, i))).*regiones_plantilla;
end
imagenF=uint8(imagenF);

% figure();
% imshow(imagenF);

%%% Se etiquetan las regiones
[A,NumCaras] = bwlabel(regiones_plantilla,8);

NumCaras

%%%Se crean matrices para guardar los resultados

etiquetas=zeros((NumCaras), 6);
resultados=zeros((NumCaras),4);

%%%%%%%% calcula coordenadas de las caras %%%%%%%%%
for a=1:NumCaras

    [m n] = size(A);

    left(a,1) = -1;
    right (a,1) = -1;
    up (a,1)=-1;
    down (a,1)= -1;

    for j=1:n,
        for i=1:m,
            if (A(i,j) == a)
                left(a,1) = j;
                break;
            end;
        end;
        if (left ~= -1) break; end;
    end;

    for j=n:-1:1,
        for i=1:m,
            if (A(i,j) == a)
                right(a,1) = j;
                break;
            end;
        end;
        if (right ~= -1) break; end;
    end;

    for i=1:m,
        for j=1:n,
            if (A(i,j) == a)
                up (a,1)= i;
                break;
            end;
        end;
        if (up ~= -1) break; end;
    end;

    for i= m:-1:1,
        for j=1:n,
            if (A(i,j) == a)
                down (a,1)= i;
                break;
            end;
        end;
    end;
end;
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
end;
if (down ~= -1) break; end;
end;

if (left(a,1) == -1) left (a,1)=1;
end;
if (right(a,1) == -1) right (a,1)= n;
end;
if (up (a,1)== -1) up(a,1)=1;
end;
if (down (a,1)== -1) down (a,1)= m;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Pinta rectangulos en las caras detectadas %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=left(a,1);
y=up(a,1);
w=right(a,1)-left(a,1);
h=down(a,1)-up(a,1);
hold on
rectangle('Position',[x,y,w,h],'LineWidth',3,'EdgeColor','r');
hold off
title('Detection final')

%%% Guarda las coordenadas de las regiones
resultados(i,:)=[x,y,w,h];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Llama a la función de etiquetado
[colorPelo, colorOjos, calvo, barba, mujer, largo]=etiquetado(imagen, x, y, w, h);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Guarda las etiquetas en una matriz
etiquetas(i,:)=[colorPelo, colorOjos, calvo, barba, mujer, largo];

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% obtenerCrCb %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Funcion que devuelve los valores cromaticos de la imagen de %entrada.
function [cr, cb] = obtenerCrCb(imagen)
im=imread(imagen);
% Convierte de RGB a YCbCr.
imycbcr = rgb2ycbcr(im);
% Matriz de filtro paso bajo.
lpf = 1/9 * ones(3);
% Vectores de los valores de cb y cr.
cr = imycbcr(:,:,3);
cb = imycbcr(:,:,2);
% Pasa el filtro de paso bajo.
cr = filter2(lpf, cr);
cb = filter2(lpf, cb);
% Concatena todas las filas.
cr = reshape(cr, 1, prod(size(cr)));
cb = reshape(cb, 1, prod(size(cb)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% histograma %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Funcion para pintar en 2D el histograma cromatico.
function histograma(cr, cb)
%Crea una matriz de ceros de 256 x 256.
hist = zeros(256);
% Redondea los valores de crominancia a enteros.
cr = round(cr);
cb = round(cb);
% Crea el histograma en 2D.
for i = 1:length(cr)
hist(cr(i),cb(i)) = hist(cr(i),cb(i))+1;
end
% Pinta el histograma
surf(hist)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% modelo_piel %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que utiliza los valores de crominancia de una base de %datos de pieles para crear
un modelo de Gaussiana en 2D.
function [crmean,cbmean,crbcov]=modelo_piel()

%Obtener los valores de crominancia de 13 imagenes de pieles.
[cr1, cb1] = obtenerCrCb('piel/1.jpg');
[cr2, cb2] = obtenerCrCb('piel/2.jpg');
[cr3, cb3] = obtenerCrCb('piel/3.jpg');
[cr4, cb4] = obtenerCrCb('piel/4.jpg');
[cr5, cb5] = obtenerCrCb('piel/5.jpg');
[cr6, cb6] = obtenerCrCb('piel/6.jpg');
```


Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
[cr7, cb7] = obtenerCrCb('piel/7.jpg');
[cr8, cb8] = obtenerCrCb('piel/8.jpg');
[cr9, cb9] = obtenerCrCb('piel/9.jpg');
[cr10, cb10] = obtenerCrCb('piel/10.jpg');
[cr11, cb11] = obtenerCrCb('piel/11.jpg');
[cr12, cb12] = obtenerCrCb('piel/12.jpg');
[cr13, cb13] = obtenerCrCb('piel/13.jpg');
[cr14, cb14] = obtenerCrCb('piel/14.jpg');
[cr15, cb15] = obtenerCrCb('piel/15.jpg');
[cr16, cb16] = obtenerCrCb('piel/16.jpg');
[cr17, cb17] = obtenerCrCb('piel/17.jpg');

%Concatena todos los valores.
cr = [cr1 cr2 cr3 cr4 cr5 cr6 cr7 cr8 cr9 cr10 cr11 cr12 cr13 cr14 cr15 cr16 cr17];
cb = [cb1 cb2 cb3 cb4 cb5 cb6 cb7 cb8 cb9 cb10 cb11 cr12 cb13 cb14 cb15 cb16 cb17];
%Calcula la media y la covarianza de dichos valores.
crmean = mean(cr);
cbmean = mean(cb);
crcbcov = cov(cr,cb);
% Se usan los valores de Cr y Cb para crear la funcion de %distribucion conjunta la cual se
asemeja a una Gaussiana.
prob = zeros(256);
for r = 0:255
    for b = 0:255
        x = [(r - crmean); (b - cbmean)];
        prob(r+1,b+1) = [power(2*pi*power(det(crcbcov),0.5),-1)]*exp(-0.5* x'*inv(crcbcov)* x);
    end
end
%%Pinta el histograma en 2D
figure();
histograma(cr,cb)
title('Histograma en 2D del Modelo Cromatico')
%Pinta la funcion de distribucion conjunta.
figure();
surf(prob)
title('Distribucion del Modelo Gaussiano')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% probabilidad %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Convierte la imagen original en una imagen de probabilidad de piel
%normalizada.
function[probabilidad_piel]=probabilidad(nombre,crmean,cbmean,cov).
img = imread(nombre);

imgycbcr = rgb2ycbcr(img);
[m,n,l] = size(img);
probabilidad_piel = zeros(m,n);
for i = 1:m
    for j = 1:n
        % Obtiene los valores de crominancia para cada pixel.
        cr = double(imgycbcr(i,j,3));
        cb = double(imgycbcr(i,j,2));
        %Calcular la probabilidad de que sea piel para cada pixel.
        x = [(cr-crmean); (cb-cbmean)];
        probabilidad_piel(i,j) = [power(2*pi*power(det(cov),0.5),-1)]*exp(-0.5* x'*inv(cov)* x);
    end
end
% Pasa un filtro paso bajo.
lpf= 1/9*ones(3);
probabilidad_piel = filter2(lpf,probabilidad_piel);
%Normaliza los valores de probabilidad respecto al valor maximo.
probabilidad_piel = probabilidad_piel./max(max(probabilidad_piel));
%Muestra la imagen de probabilidad de piel en escala de gris.
figure();
subplot(3,3,1);
imshow(img, [0 1])
title('Imagen RGB original')
subplot(3,3,2);
imshow(probabilidad_piel, [0 1])
title('Imagen de Probabilidad de Piel')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% segmentacion %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que coge la imagen de probabilidad de piel como entrada y %%la segmenta en una
imagen binaria usando un umbral adaptativo.
function [piel_segmentada,umbral_optimo] = segmentacion(probabilidad_piel)

[m,n] = size(probabilidad_piel);
temp = zeros(m,n);
diff_list = [];
% Establece un rango de umbral y un tamaño de paso mediante la %experiencia.
alto=0.55;
bajo=0.01;
paso=-0.1;
factor=1;
indx_count=[(alto-bajo)/abs(paso)]+2;
% Busqueda de el umbral optimo.
for umbral = alto:paso:bajo
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
piel_segmentada = zeros(m,n);
piel_segmentada(find(probabilidad_piel>umbral)) = 1;
diff = sum(sum(piel_segmentada - temp));
diff_list = [diff_list diff];
temp = piel_segmentada;
end
[C, indx] = min(diff_list);
umbral_optimo = (indx_count-indx)*abs(paso)*factor;
% Resetea la imagen con ese umbral optimo.
piel_segmentada = zeros(m,n);
piel_segmentada(find(probabilidad_piel>umbral_optimo)) = 1;

subplot(3,3,3);
imshow(piel_segmentada, [0 1])
title('Imagen segmentada de piel')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%proceso_regiones %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que aplica operadores morfologicos y etiqueta las %regiones.
function[etiqueta,num]=proceso_regiones(piel_segmentada)
[m,n] = size(piel_segmentada);

% se0 = strel('disk',2);
% erodedBW0=zeros(m,n);
% erodedBW0= imerode(binary_skin,se0);

sin_agujeros=zeros(m,n);
sin_agujeros = imfill(piel_segmentada,'holes');

se2 = strel('disk',10);
erosion=zeros(m,n);
erosion = imerode(sin_agujeros,se2);
subplot(3,3,4);
imshow(erosion)
% title({'Despues de la Erosion';'(disk size: 10)'})

se1 = strel('disk',8);
dilatacion=zeros(m,n);
dilatacion=imdilate(erosion,se1);
% Multiplica la imagen dilatada con la imagen segmentada de piel %para mantener los agujeros
y luego pinta dicha imagen.
dilatacion = immultiply(dilatacion,piel_segmentada);
subplot(3,3,5);
imshow(dilatacion)
title({'Despues de la Dilation';'(disk size: 8)'})
% Etiqueta regiones de piel.
etiqueta=zeros(m,n);
[etiqueta,num] = bwlabel(dilatacion,8);
if(num==0)
etiqueta=0;
num=0;
else
% Pinta el fondo de las regiones etiquetadas en negro.
color_regiones=zeros(m,n);
color_regiones= label2rgb(etiqueta, 'hsv', 'black', 'shuffle');
% Muestra las regiones etiquetadas coloreadas de la imagen.
subplot(3,3,6);
imshow(color_regiones)
title({'Regiones etiquetadas';['(',num2str(num),' regiones)']})

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% euler %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que selecciona solo las regiones que tienen al menos un %agujero.
function [regiones_euler, num]=euler(etiquetas, num)

if(num==0)
regiones_euler=0;
num=0;
else
% Calcula el numero de euler para cada region.
e = regionprops(etiquetas,'EulerNumber');
euler=cat(1,e.EulerNumber);
% Calcula el numero de agujeros de cada region.
agujeros=1-euler;
% Coge las regiones que tiene al menos un orificio.
region_index = find(agujeros>=1);
[m,n]=size(etiquetas);
regiones_euler=zeros(m,n);
% Crea una nueva imagen binaria solo con las regiones que pasan el %test de eurler.
for i=1:length(region_index)
% Calcula las coordenadas de cada region.
[x,y] = find(bwlabel(etiquetas) == region_index(i));
% Obtiene una imagen que tiene solo esa region, el resto se pone en
% negro.
region = bwselect(etiquetas,y,x,8);
regiones_euler=regiones_euler+region;
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
end
if (length(region_index)==0)
num=0;
regiones_euler=0;
end
% Pinta la imagen solo con las regiones que han pasado el test de %euler.
subplot(3,3,7);
imshow(regiones_euler)
title({'Despues del test de Euler';['(',num2str(length(region_index)), ' regiones']})
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ratio %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que determina si la relacion de aspecto esta dentro del %rango permitido para ser
una region de cara.
function [regiones_ratio, num]=ratio(regiones_euler, num)
if(num==0)
regiones_ratio=0;
num=0;
else
[m,n]=size(regiones_euler);
% Rellena los agujeros de la imagen.
sin_agujeros = imfill(regiones_euler,'holes');

sel = strel('disk',3);
dilatacion=zeros(m,n);
dilatacion=imdilate(sin_agujeros,sel);
% Etiqueta las regiones.
[etiquetas,num] = bwlabel(dilatacion,8);
[ratio]=obtener_ratio(etiquetas);
% Coge las regions que tienen una relacion de aspecto que esta %dentro del rango permitido.
region_index = find(ratio>=1 & ratio<=3.5);
regiones_ratio=zeros(m,n);
% Crea una imagen binaria solo con las regiones que pasen el test %de ratio.
for i=1:length(region_index)
% Calcula las coordenadas para cada region.
[x,y] = find(bwlabel(sin_agujeros) == region_index(i));
% Obtiene una imagen que tiene solo esa region, el resto se pone en
% negro.
region = bwselect(sin_agujeros,y,x,8);
regiones_ratio=regiones_ratio+region;
end
if (length(region_index)==0)
num=0;
regiones_euler=0;
end
% Pinta la imagen solo con las regiones que han pasado el test de %ratio.
subplot(3,3,8);
imshow(regiones_ratio)
title({'Despues de pasar el Test de Ratio';['(',num2str(length(region_index)), '
regiones']})
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plantilla %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que devuelve una imagen con las regiones que pasan el
%test de plantilla coincidente.
function [regiones_plantilla, RectCoord, num]=plantilla(regiones_ratio, imagenRGB,
plantilla, num)

if(num==0)
regiones_plantilla=0;
RectCoord=0;
num=0;
else
imgray=rgb2gray(imread(imagenRGB));
plantilla=imread(plantilla);

[etiquetas,num] = bwlabel(regiones_ratio,8);
[m,n]=size(regiones_ratio);

% Calcula el angulo de orientacion para cada region.
orientacion = regionprops(etiquetas,'Orientation');
angulos=cat(1,orientacion.Orientation);

% Calcula el centroide de cada region.
c = regionprops(etiquetas,'Centroid');
centro=cat(1,c.Centroid);

% Imagen con las regiones que pasan el test de plantilla.
regiones_plantilla=zeros(m,n);
gray_matched=zeros(m,n);
% Redimensiona, rota y corta la plantilla de acuerdo a las %propiedades de la region.
for j=1:num,
% Calcula las coordenadas de cada region.
[x,y] = find(etiquetas == j);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
% Obtiene una imagen solo con esta region y el fondo puesto a %negro.
region = bwselect(regiones_ratio,y,x,8);
% Genera una imagen en escala de gris con solo una region de cara.
una_cara=immultiply(region,imgray);
% Calcula los centroides de cada region.
cx1=centro(j,1);
cy1=centro(j,2);
% Saca la altura y la anchura de cada region.
p=regionprops(bwlabel(region),'BoundingBox');
boxdim=cat(1,p.BoundingBox);
regw=boxdim(3);
regh=boxdim(4);
ratio=regh/regw;
% Si la region es muy larga, establece una nueva altura y un nuevo
% centroide razonable.
if(ratio>1.6)
    regh=1.5*regw;
    cy1=cy1-(0.1*regh);
end
% Redimensiona la plantilla en la misma escala que la region.
modelo_resize=imresize(plantilla,[regh regw],'bilinear');
% Rota la plantilla redimensionada con el angulo de orientacion de %la region.
if(angulos(j)>0)
    modelo_rotado=imrotate(modelo_resize,angulos(j)-90,'bilinear','loose');
else
    modelo_rotado=imrotate(modelo_resize,90+angulos(j),'bilinear','loose');
end
% Calcula el centroide y el tamaño de la plantilla.
modelo=im2bw(modelo_rotado,0);
[g,h]=size(modelo);
% Garantiza que la region de la plantilla tiene solo una region.
nuevoModelo = bwmorph(modelo,'clean');
[L,no]=bwlabel(nuevoModelo,8);
if(no==1)
    solo=nuevoModelo;
else
    ar=regionprops(bwlabel(nuevoModelo),'Area');
    areas=cat(1,ar.Area);
    [C,I]=max(areas);
    % Calcula las coordenadas para cada region.
    [x1,y1] = find(bwlabel(nuevoModelo)== I);
    % Obtiene una imagen que solo tiene esa region y el resto se pone %en negro.
    solo = bwselect(nuevoModelo,y1,x1,8);
end
% Rellena la plantilla y cortamos.
modeloSinAgujeros=regionprops(bwlabel(solo),'FilledImage');
recortado=modeloSinAgujeros.FilledImage;
% Saca el tamaño de la plantilla rotada y escala, despue de %cortala.
[modh,modw]=size(recortado);
% Corta la plantilla en escala de gris con el mismo tamaño que la %plantilla de las
% etiquetas.
modelo_recortado=imresize(modelo_rotado,[modh modw],'bilinear');
% Calcula el centroide del modelo escalado, rotado y cortado.
centroide=regionprops(bwlabel(recortado),'Centroid');
centroides=cat(1,centroide.Centroid);
cx2=centroides(1,1);
cy2=centroides(1,2);

mfit = zeros(size(una_cara));
mfitbw = zeros(size(una_cara));
[limy, limx] = size(mfit);

% Calcula las coordeadas de donde la cara de la plantilla va a estar en la
% imagen principal.

startx = cx1-cx2;
starty = cy1-cy2;
endx = startx + modw-1;
endy = starty + modh-1;

% Comprueba los limites de la imagen.
startx = comprueba_limites(startx,limx);
starty = comprueba_limites(starty,limy);
endx = comprueba_limites(endx,limx);
endy = comprueba_limites(endy,limy);

% Genera una nueva imagen que tiene el mismo tamaño que la %original, pero con la cara de
% la plantilla en ella.

for i=starty:endy,
    for j=startx:endx,
        mfit(round(i),round(j)) = modelo_recortado(round(i-starty+1),round(j-startx+1));
    end;
end;

% Obtiene la correlacion entre la plantilla y la region.
gray_matched=gray_matched+mfit;
correlacion =corr2(mfit,una_cara);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
% Si la correlacion es mayor que un umbral, añadimos esa region a %la lista de caras.
if(correlacion>=0.6)
regiones_plantilla=regiones_plantilla+region;
end;

% Pinta la imagen solo con las regiones que han pasado el test de %plantilla coincidente.
subplot(3,3,9);
imshow(gray_matched,[0 255])
title('Template Matching')

%%Obtener las coordenadas de las regiones.
[l,r,u,d] = resize(regiones_plantilla);

sx = startx+l;
sy = starty+u;
RectCoord = [sx sy (r-l) (d-u)];

end;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% obtener_ratio %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funcion que devuelve la relacion de aspecto de las regiones de la %imagen binaria.
function [lista_ratio] = obtener_ratio(region)
% Calcula la longitud del eje mayor(de la elipse) para cada region.
ejeMayor = regionprops(region,'MajorAxisLength');
mayor_length=cat(1,ejeMayor.MajorAxisLength);
%Calcula la longitud del eje menor(de la elipse) para cada region.
ejeMenor = regionprops(region,'MinorAxisLength');
menor_length=cat(1,ejeMenor.MinorAxisLength);
% Calcula la relacion de aspecto.
lista_ratio=mayor_length./menor_length;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% comprueba limites %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Verifica que las coordenadas estan dentro de la region de la %imagen.
function newcoord = comprueba_limites(coord,maxval)

newcoord = coord;
if (newcoord<1) newcoord=1; end;
if (newcoord>maxval) newcoord=maxval; end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- Función ***etiquetado, raza, Ojos, pelo, sexo, obtenerHSVPElo, obtenerHSVPiel***

```
function [colorPelo, colorOjos, calvo, barba, mujer, largo] = etiquetado(imagen, x, y, w, h)

I=imread(imagen);

[colorRaza] = raza(imagen, x, y, w, h);
[colorOjos] = Ojos(imagen, x, y, w, h);
[colorPelo, calvo, barba, largo] = pelo(imagen, x, y, w, h, colorRaza);
[mujer]=sexo(imagen, calvo, barba, largo);

% colorPelo
% calvo
% barba
% mujer
% largo
% colorOjos

function [colorRaza] = raza(imagen, x, y, w, h)

img=imread(imagen);
figure();
imshow(img);

R=img(:,:,1);
G=img(:,:,2);
B=img(:,:,3);

%Pinta el rectangulo de la cara
hold on;
rectangle('Position',[x,y,w,h], 'LineWidth',3, 'EdgeColor','r');
hold off;
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
%%%Recorta la cara
I2 = imcrop(img,[x,y,(w-1),(h-1)]);

%%% Recorta el trozo de frente
I3=imcrop(I2,[ (2*w/5),(h/5),(w/8),(w/8)]);

figure()
imshow(I3)

colorRaza=1;

I3=rgb2ycbcr(I3);

Y=I3(:,:,1);
CB=I3(:,:,2);
CR=I3(:,:,3);

media=mean(mean(Y));

figure();
imhist(Y);
title('y');
figure()
imhist(CB);
figure()
imhist(CR);

condicionNegro= (media<145);
condicionBlanco= (media>=145);

if condicionNegro

    disp('es negro')
    colorRaza=0;

elseif condicionBlanco

    disp('es blanco')

end

function [colorOjos]= Ojos(imagen, x, y, w, h)
img=imread(imagen);

I=double(img);
imshow(img)

H=size(I,1);
W=size(I,2);

R=I(:,:,1);
G=I(:,:,2);
B=I(:,:,3);

%%Pinta rectangulo cara
hold on;
rectangle('Position',[x,y,w,h],'LineWidth',3,'EdgeColor','r');
hold off;

%%%Recortamos la cara

I2 = imcrop(img,[x,y,(w-1),(h-1)]);

% I2=double(I2);
R=I2(:,:,1);
G=I2(:,:,2);
B=I2(:,:,3);

%%%%%%%% Filtro de mediana %%%%%%%%%

filtrada(:,:,1) =medfilt2(R);
filtrada(:,:,2) =medfilt2(G);
filtrada(:,:,3) =medfilt2(B);

%%%%%%%% Segmentacion piel RGB %%%%%%%%%
maxi=max(filtrada,[],3); % Maximo en los 3 colores
mini=min(filtrada,[],3); % Mínimo en los 3 colores
%% Se obtiene a partir de una imagen color-RGB una imagen binaria donde los
% valores 1 corresponden a la presencia de píxeles de piel.

% Condiciones
c1=(R>95) & (G>40) & (B>20) & ((maxi-mini)>15);
c2=abs(R-G)>15;
c3=(R>G) & (R>B);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
% Detección píxeles de piel
piel=(c1&c2&c3);

figure();
imshow(piel)
title('Segmentacion piel');

inversa=~piel;

figure();
imshow(inversa)
title('inversa');

%% Etiquetado de regiones
[regiones,num] = bwlabel(inversa,8);

%% Propiedades de la region
propiedades=regionprops(regiones,'BoundingBox','MinorAxisLength','MajorAxisLength','Area','Image');
numCaras=0;
detectada=[];

for i = 1:num
    region=(regiones==i); %Selección de una región
    %Calculo del perimetro, relación de aspecto y factor de forma
    perimetro= bwperim(region);
    area=bwarea(perimetro);
    relacion_aspecto(i)=propiedades(i).MinorAxisLength/propiedades(i).MajorAxisLength;
    factor_forma(i)=(4*pi)*(propiedades(i).Area/area^2);

    %Condiciones para que sea ojo
    umbral=(w/3);
    condicion=(factor_forma(i)>0.2) & (propiedades(i).Area>umbral);
    if condicion
        numCaras=numCaras+1;
        detectada(numCaras).image=propiedades(i).Image;
        detectada(numCaras).posicion=propiedades(i).BoundingBox;
    end
end

numCaras
figure();
imshow(inversa);

correlacionNegro=zeros(length(detectada),3);
correlacionMarron=zeros(length(detectada),3);
correlacionAzul=zeros(length(detectada),3);
correlacionVerde=zeros(length(detectada),3);
suma=zeros(length(detectada),4);

for i=1:length(detectada)

    coordenadas = round((detectada(i).posicion));
    A=detectada(i).image;

    %%% Coordenadas del candidato a ojo
    x=coordenadas(1);
    y=coordenadas(2);
    w=coordenadas(3);
    h=coordenadas(4);

    hold on
    rectangle('Position',[x,y,w,h], 'LineWidth',3, 'EdgeColor','r');
    hold off

    %%% Recorta el candidato a ojo
    I3 = imcrop(I2,[x,y,(w-1),(h-1)]);

    figure();
    imshow(I3);
    title('Candidato a ojo');

    I3R=I3(:, :,1);
    I3G=I3(:, :,2);
    I3B=I3(:, :,3);

    %%%plantilla ojoNegro
    negro=imread('ojoNegro.jpg');
    plantillaNegro=imresize(negro,[h w]);
    %figure();
    %imshow(plantillaNegro);

    plantillaNegroR=plantillaNegro(:, :,1);
    plantillaNegroG=plantillaNegro(:, :,2);
    plantillaNegroB=plantillaNegro(:, :,3);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
%% Correlacion del candidato a ojo con las plantillas
correlacionNegroR =corr2(I3R, plantillaNegroR);
correlacionNegroG =corr2(I3G, plantillaNegroG);
correlacionNegroB =corr2(I3B, plantillaNegroB);

correlacionNegro(i,:)=[correlacionNegroR,correlacionNegroG,correlacionNegroB];

N=correlacionNegroR+correlacionNegroG+correlacionNegroB;

%%plantilla ojoMarron
marron=imread('ojoMarron.jpg');
plantillaMarron=imresize(marron,[h w]);
% figure();
% imshow(plantillaMarron);

plantillaMarronR=plantillaMarron(:,:,1);
plantillaMarronG=plantillaMarron(:,:,2);
plantillaMarronB=plantillaMarron(:,:,3);

%% Correlacion del candidato a ojo con las plantillas
correlacionMarronR =corr2(I3R, plantillaMarronR);
correlacionMarronG =corr2(I3G, plantillaMarronG);
correlacionMarronB =corr2(I3B, plantillaMarronB);

correlacionMarron(i,:)=[correlacionMarronR,correlacionMarronG,correlacionMarronB];
M=correlacionMarronR+correlacionMarronG+correlacionMarronB;

%%plantilla ojoAzul
azul=imread('ojoAzul.jpg');
plantillaAzul=imresize(azul,[h w]);
% figure();
% imshow(plantillaAzul);

plantillaAzulR=plantillaAzul(:,:,1);
plantillaAzulG=plantillaAzul(:,:,2);
plantillaAzulB=plantillaAzul(:,:,3);

%% Correlacion del candidato a ojo con las plantillas
correlacionAzulR =corr2(I3R, plantillaAzulR);
correlacionAzulG =corr2(I3G, plantillaAzulG);
correlacionAzulB =corr2(I3B, plantillaAzulB);

correlacionAzul(i,:)=[correlacionAzulR,correlacionAzulG,correlacionAzulB];
A=correlacionAzulR+correlacionAzulG+correlacionAzulB;

%%plantilla ojoVerde
verde=imread('ojoVerde.jpg');
plantillaVerde=imresize(verde,[h w]);
% figure();
% imshow(plantillaVerde);

plantillaVerdeR=plantillaVerde(:,:,1);
plantillaVerdeG=plantillaVerde(:,:,2);
plantillaVerdeB=plantillaVerde(:,:,3);

%% Correlacion del candidato a ojo con las plantillas
correlacionVerdeR =corr2(I3R, plantillaVerdeR);
correlacionVerdeG =corr2(I3G, plantillaVerdeG);
correlacionVerdeB =corr2(I3B, plantillaVerdeB);

correlacionVerde(i,:)=[correlacionVerdeR,correlacionVerdeG,correlacionVerdeB];
V=correlacionVerdeR+correlacionVerdeG+correlacionVerdeB;

suma(i,:)=[N,M,A,V];

end

%%Recorremos la matriz para ver cual es el valor mayor, y %guardamos la columna del valor
mayor
z=0;
valor=0;
for j=1:4
for k=1:length(detectada)

    if suma(k,j)>z
        z=suma(k,j);
        valor=j;
    end
end

end
```


Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
end

colorOjos=1;
if(valor==1 & z>0.90)
    disp('Tiene los ojos negros')
elseif (valor==2 & z>0.90)
    disp('Tiene los ojos marrones')
elseif (valor==3 & z>0.90)
    disp('Tiene los ojos azules')
elseif (valor==4 & z>0.90)
    disp('Tiene los ojos verdes')
else (z<0.90)
    disp('Problema en el etiquetado')
    colorOjos=0;
end

function [colorPelo, calvo, barba, largo]=pelo(imagen, x, y, w, h, colorRaza)

I=imread(imagen);

figure();
imshow(I);
title('pelo');

%Pinta rectangulo cara
hold on;
rectangle('Position',[x,y,w,h], 'LineWidth',3, 'EdgeColor','r');
hold off;

%%Recorta el pelo
X=x;
Y= y -(0.25*h);
W=w;
H=0.25*h;

I2 = imcrop(I,[X Y W H]);
figure();
imshow(I2);

%% Se obtienen los valores medios de HSV del pelo recortado
HSV = rgb2hsv(I2);

a=HSV(:, :,1);
H=mean(mean(a));
b=HSV(:, :,2);
S=mean(mean(b));
c=HSV(:, :,3);
V=mean(mean(c));

%%Recortamos la barba
A=x+(0.2)*w;
B= y+(0.8)*h;
C=(0.55)*w;
D=(1/9)*h;

figure()
imshow(I);

%Pinta el rectangulo de la barba en la cara
hold on;
rectangle('Position',[A,B,C,D], 'LineWidth',3, 'EdgeColor','r');
hold off;

I3 = imcrop(I,[A B C D]);
figure();
imshow(I3);

%% sacamos valores medios de HSV del la barbilla recortada
HSV = rgb2hsv(I3);

a=HSV(:, :,1);
Hb=mean(mean(a));
b=HSV(:, :,2);
Sb=mean(mean(b));
c=HSV(:, :,3);
Vb=mean(mean(c));

%%%% Obtener Valores de pelo de HSV maximo y minimo en funcion %de unas plantillas.

[HmaxM,HminM, SmaxM, SminM, VmaxM,
VminM]=obtenerHSVPelo(('pelo/moreno1.jpg'),('pelo/moreno2.jpg'),('pelo/moreno3.jpg'),('pelo/
moreno4.jpg'),('pelo/moreno5.jpg'),('pelo/moreno6.jpg'),('pelo/moreno7.jpg'),('pelo/moreno8.
jpg'));
condicionMoreno= (HminM<H) && (H<HmaxM) && (SminM<S) && (S<SmaxM) && (VminM<V) && (V<VmaxM);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
[HmaxR, HminR, SmaxR, SminR, VmaxR, VminR]=
obtenerHSVPelo(('pelo/rubio1.jpg'),('pelo/rubio2.jpg'),('pelo/rubio3.jpg'),('pelo/rubio4.jpg',
'),('pelo/rubio5.jpg'),('pelo/rubio6.jpg'),('pelo/rubio7.jpg'),('pelo/rubio8.jpg'));
condicionRubio= (HminR<H) && (H<HmaxR) && (SminR<S) && (S<SmaxR) && (VminR<V) && (V<VmaxR);

[HmaxC, HminC, SmaxC, SminC, VmaxC, VminC]=
obtenerHSVPelo(('pelo/castano1.jpg'),('pelo/castano2.jpg'),('pelo/castano3.jpg'),('pelo/cast
ano4.jpg'),('pelo/castano5.jpg'),('pelo/castano6.jpg'),('pelo/castano7.jpg'),('pelo/castano8
.jpg'));
condicionCastano= (HminC<H) && (H<HmaxC) && (SminC<S) && (S<SmaxC) && (VminC<V) &&
(V<VmaxC);

[HmaxP,HminP, SmaxP, SminP, VmaxP, VminP]=
obtenerHSVPelo(('pelo/pelirrojo1.jpg'),('pelo/pelirrojo2.jpg'),('pelo/pelirrojo3.jpg'),('pel
o/pelirrojo4.jpg'),('pelo/pelirrojo5.jpg'),('pelo/pelirrojo6.jpg'),('pelo/pelirrojo7.jpg'),('
'pelo/pelirrojo8.jpg'));
condicionPelirrojo= (HminP<H) && (H<HmaxP) && (SminP<S) && (S<SmaxP) && (VminP<V) &&
(V<VmaxP);

[HmaxB,HminB, SmaxB, SminB, VmaxB, VminB]=
obtenerHSVPiel(('piel/1.jpg'),('piel/2.jpg'),('piel/3.jpg'),('piel/4.jpg'),('piel/5.jpg'),('
piel/6.jpg'),('piel/7.jpg'),('piel/8.jpg'),('piel/9.jpg'),('piel/10.jpg'),('piel/11.jpg'),('
piel/12.jpg'),('piel/13.jpg'),('piel/14.jpg'),('piel/15.jpg'),('piel/16.jpg'),('piel/17.jpg'
));
condicionPiel= (HminB<H) && (H<HmaxB) && (SminB<S) && (S<SmaxB) && (VminB<V) && (V<VmaxB);

colorPelo=1;
calvo=0;
barba=1;
largo=0;

%% Condiciones de color de pelo
if condicionRubio
    disp('Tiene el pelo rubio');

elseif condicionCastano
    disp('Tiene el pelo Castaño');

    elseif condicionPelirrojo
        disp('Tiene el pelo pelirrojo');

    elseif condicionMoreno
        disp('Tiene el pelo moreno');

    elseif condicionPiel
        disp('Esta calvo');
        calvo=1;
        end

%%% Condiciones barba con los mismos valores de las plantillas.

condicionMoreno= (HminM<Hb) && (Hb<HmaxM) && (SminM<Sb) && (Sb<SmaxM) && (VminM<Vb) &&
(Vb<VmaxM);

condicionRubio= (HminR<Hb) && (Hb<HmaxR) && (SminR<Sb) && (Sb<SmaxR) && (VminR<Vb) &&
(Vb<VmaxR);

condicionCastano= (HminC<Hb) && (Hb<HmaxC) && (SminC<Sb) && (Sb<SmaxC) && (VminC<Vb) &&
(Vb<VmaxC);

condicionPelirrojo= (HminP<Hb) && (Hb<HmaxP) && (SminP<Sb) && (Sb<SmaxP) && (VminP<Vb) &&
(Vb<VmaxP);

condicionPiel=(HminB<=Hb) && (Hb<=HmaxB) && (SminB<=Sb) && (Sb<=SmaxB) && (VminB<=Vb) &&
(Vb<=VmaxB);

if (colorRaza==1)

if condicionRubio
    disp('Tiene barba');

elseif condicionCastano
    disp('Tiene barba');

    elseif condicionPelirrojo
        disp('Tiene barba');

    elseif condicionMoreno
        disp('tiene barba');
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
else
    if condicionPiel
        disp('No tiene barba');
        barba=0;
    else
        disp('tiene barba');
    end
end

else
    disp('no tiene barba')
    barba=0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure();
imshow(I);

%%%Recorta la region del lado izquierdo del cuello
Y=y+(0.75*h);
X= x -(0.25*w);
W=(0.25*w);
H=(0.5*h);

%Pinta el rectangulo de la zona del pelo largo en la cara
hold on;
rectangle('Position',[X,Y,W,H],'LineWidth',3,'EdgeColor','r');
hold off;

I4 = imcrop(I,[X Y W H]);
figure();
imshow(I4);
title('Región lado izquierdo');

%% Se obtienen los valores medios de HSV del la barbilla %%%recortada
HSV = rgb2hsv(I4);

HI=HSV(:,:,1);
HI=mean(mean(HI));
SI=HSV(:,:,2);
SI=mean(mean(SI));
VI=HSV(:,:,3);
VI=mean(mean(VI));

condicionMoreno= (HminM<HI) && (HI<HmaxM) && (SminM<SI) && (SI<SmaxM) && (VminM<VI) &&
(VI<VmaxM);
condicionRubio= (HminR<HI) && (HI<HmaxR) && (SminR<SI) && (SI<SmaxR) && (VminR<VI) &&
(VI<VmaxR);
condicionPelirrojo= (HminP<HI) && (HI<HmaxP) && (SminP<SI) && (SI<SmaxP) && (VminP<VI) &&
(VI<VmaxP);
condicionCastano= (HminC<HI) && (HI<HmaxC) && (SminC<SI) && (SI<SmaxC) && (VminC<VI) &&
(VI<VmaxC);

if condicionRubio
    largo=1;
elseif condicionCastano
    largo=1;
    elseif condicionPelirrojo
        largo=1;
    elseif condicionMoreno
        largo=1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%Recorta la región derecha del cuello
Y=y+(0.75*h);
X=(x+w);
W=(0.25*w);
H=(0.5*h);

figure();
imshow(I);

%Pinta el rectangulo de la zona del pelo largo en la cara
hold on;
rectangle('Position',[X,Y,W,H],'LineWidth',3,'EdgeColor','r');
hold off;

I5 = imcrop(I,[X Y W H]);
figure();
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
imshow(I5);
title('Region lado derecho');

%% Se obtienen los valores medios de HSV del la barbilla %recortada
HSV = rgb2hsv(I5);

HD=HSV(:,:,1);
HD=mean(mean(HD));
SD=HSV(:,:,2);
SD=mean(mean(SD));
VD=HSV(:,:,3);
VD=mean(mean(VD));

condicionMoreno= (HminM<HD) && (HD<HmaxM) && (SminM<SD) && (SD<SmaxM) && (VminM<VD) &&
(VD<VmaxM);
condicionRubio= (HminR<HD) && (HD<HmaxR) && (SminR<SD) && (SD<SmaxR) && (VminR<VD) &&
(VD<VmaxR);
condicionPelirrojo= (HminP<HD) && (HD<HmaxP) && (SminP<SD) && (SD<SmaxP) && (VminP<VD) &&
(VD<VmaxP);
condicionCastano= (HminC<HD) && (HD<HmaxC) && (SminC<SD) && (SD<SmaxC) && (VminC<VD) &&
(VD<VmaxC);

if condicionRubio
    largo=1;

elseif condicionCastano
    largo=1;

    elseif condicionPelirrojo
        largo=1;

    elseif condicionMoreno
        largo=1;
end

function[mujer]=sexo(imagen, calvo, barba, largo)

img=imread(imagen);
I=double(img);

imshow(img);

H=size(I,1);
W=size(I,2);

R=I(:,:,1);
G=I(:,:,2);
B=I(:,:,3);

%%%%%%%%%% Filtro de mediana %%%%%%%%%%%

filtrada(:,:,1) =medfilt2(R);
filtrada(:,:,2) =medfilt2(G);
filtrada(:,:,3) =medfilt2(B);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Segmentacion piel RGB %%%%%%%%%%%
maxi=max(filtrada,[],3); % Maximo en los 3 colores
mini=min(filtrada,[],3); % Mínimo en los 3 colores
%% Se obtiene a partir de una imagen color-RGB una imagen %%binaria donde los valores 1
corresponden a la presencia de pixeles de piel.

% Condiciones
c1=(R>95) & (G>40) & (B>20) & ((maxi-mini)>15);
c2=abs(R-G)>15;
c3=(R>G) & (R>B);
% Detección pixeles de piel
piel=(c1&c2&c3);

figure();
imshow(piel)
title('Segmentacion piel');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Erosion con un disco de radio 3 %%%%%%%%%%%
ee=strel('disk',3);
erosionada=imerode(piel, ee);
figure();
imshow(erosionada);
title('Imagen erosionada');

%%%%%%%%%% Dilatacion con un disco de ratio 3 %%%%%%%%%%%

eel=strel('disk',3);
dilatada=zeros(H,W);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
dilatada=imdilate(erosionada, ee1);

%%Multiplica la imagen dilatada con la imagen segmentada
%%de piel para mantener los agujeros y luego pinta dicha imagen.
dilatada=immultiply(dilatada,piel);
figure();
imshow(dilatada);
title('Imagen dilatada');

% Etiquetado de las regiones
[imagen_etiquetada,num]=bwlabel(dilatada);

%%% Propiedades de la region
propiedades=regionprops(imagen_etiquetada,'BoundingBox','MinorAxisLength','MajorAxisLength',
'Area','EulerNumber','Solidity','Image','Centroid');
numCaras=0;
detectada=[];

datos=zeros(num,2);
%% Condiciones para ser cara
for i = 1:num
region=(imagen_etiquetada==i); %Selección de una región
%Calculo del perimetro, relación de aspecto y factor de forma
perimetro= bwperim(region);
area=bwarea(perimetro);
relacion_aspecto(i)=propiedades(i).MinorAxisLength/propiedades(i).MajorAxisLength;
factor_forma(i)=(4*pi)*(propiedades(i).Area/area^2);
euler=propiedades(i).EulerNumber;
agujeros=1-euler;
%Condiciones
condicion=(factor_forma(i)>0.1) & (relacion_aspecto(i)>0.3) & (propiedades(i).Solidity>0.7) & (agujeros>=1);

if condicion
datos(i,:)=[factor_forma(i),relacion_aspecto(i)];
numCaras=numCaras+1;
detectada(numCaras).image=propiedades(i).Image;
detectada(numCaras).posicion=propiedades(i).BoundingBox;
end

end

factorSexo=0;

for l=1:num
if (datos(l,1)<0.2 & datos(l,1)~=0 & datos(l,2)<0.7 & datos(l,2)~=0)

factorSexo=1;

end
end

numCaras
mujer
figure();
imshow(img);
for i=1:length(detectada)

coordenadas = round((detectada(i).posicion));
A=detectada(i).image;

%%% Se obtienen las coordenadas de la region de interes
x=coordenadas(1);
y=coordenadas(2);
w=coordenadas(3);
h=coordenadas(4);

hold on
rectangle('Position',[x,y,w,h], 'LineWidth',3, 'EdgeColor','r');
hold off

end

%%% Condiciones de etiquetado sexo. Tabla de verdad
condicion1=(calvo==0) & (barba==0) & (factorSexo==0) & (largo==0);
condicion2=(calvo==0) & (barba==0) & (factorSexo==0) & (largo==1);
condicion3=(calvo==0) & (barba==0) & (factorSexo==1) & (largo==0);
condicion4=(calvo==0) & (barba==0) & (factorSexo==1) & (largo==1);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
condicion5=(calvo==0) & (barba==1) & (factorSexo==0) & (largo==0);
condicion6=(calvo==0) & (barba==1) & (factorSexo==0) & (largo==1);
condicion7=(calvo==0) & (barba==1) & (factorSexo==1) & (largo==0);
condicion8=(calvo==0) & (barba==1) & (factorSexo==1) & (largo==1);

condicion9=(calvo==1) & (barba==0) & (factorSexo==0) & (largo==0);
condicion10=(calvo==1) & (barba==0) & (factorSexo==0) & (largo==1);
condicion11=(calvo==1) & (barba==0) & (factorSexo==1) & (largo==0);
condicion12=(calvo==1) & (barba==0) & (factorSexo==1) & (largo==1);

condicion13=(calvo==1) & (barba==1) & (factorSexo==0) & (largo==0);
condicion14=(calvo==1) & (barba==1) & (factorSexo==0) & (largo==1);
condicion15=(calvo==1) & (barba==1) & (factorSexo==1) & (largo==0);
condicion16=(calvo==1) & (barba==1) & (factorSexo==1) & (largo==1);

if condicion1
    disp('es un hombre');

elseif condicion2
    disp('probablemente es una mujer');

elseif condicion3
    disp('Probablemente es una mujer');

elseif condicion4
    disp('es una mujer');

elseif condicion5
    disp('es un hombre');

elseif condicion6
    disp('Probablemente un hombre');

elseif condicion7
    disp('Probablemente un hombre');

elseif condicion8
    disp('Problema en el etiquetado');

elseif condicion9
    disp('es un hombre');

elseif condicion10
    disp('mal etiquetado');

elseif condicion11
    disp('Probablemente sea un hombre');

elseif condicion12
    disp('mal etiquetado');

elseif condicion13
    disp('es un hombre');

elseif condicion14
    disp('hombre');

elseif condicion15
    disp('Mal etiquetado');

elseif condicion16
    disp('Mal etiquetado');

end

function [Hmax,Hmin,Smax,Smin,Vmax,Vmin]=
color(pelo1,pelo2,pelo3,pelo4,pelo5,pelo6,pelo7,pelo8);

%Carga y convierte a HSV las plantillas de pelo.
pelo1=imread(pelo1);
HSV = rgb2hsv(pelo1);
H1=HSV(:, :, 1);
S1=HSV(:, :, 2);
V1=HSV(:, :, 3);

Hmin1=min(H1);
Smin1=min(S1);
Vmin1=min(V1);

Hmin1=mean(Hmin1);
Smin1=mean(Smin1);
Vmin1=mean(Vmin1);

Hmax1=max(H1);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
Smax1=max(S1);
Vmax1=max(V1);

Hmax1=mean(Hmax1);
Smax1=mean(Smax1);
Vmax1=mean(Vmax1);

pelo2=imread(pelo2);
HSV = rgb2hsv(pelo2);
H2=HSV(:,:,1);
S2=HSV(:,:,2);
V2=HSV(:,:,3);

Hmin2=min(H2);
Smin2=min(S2);
Vmin2=min(V2);

Hmin2=mean(Hmin2);
Smin2=mean(Smin2);
Vmin2=mean(Vmin2);

Hmax2=max(H2);
Smax2=max(S2);
Vmax2=max(V2);

Hmax2=mean(Hmax2);
Smax2=mean(Smax2);
Vmax2=mean(Vmax2);

pelo3=imread(pelo3);
HSV = rgb2hsv(pelo3);
H3=HSV(:,:,1);
S3=HSV(:,:,2);
V3=HSV(:,:,3);

Hmin3=min(H3);
Smin3=min(S3);
Vmin3=min(V3);

Hmin3=mean(Hmin3);
Smin3=mean(Smin3);
Vmin3=mean(Vmin3);

Hmax3=max(H3);
Smax3=max(S3);
Vmax3=max(V3);

Hmax3=mean(Hmax3);
Smax3=mean(Smax3);
Vmax3=mean(Vmax3);

pelo4=imread(pelo4);
HSV = rgb2hsv(pelo4);
H4=HSV(:,:,1);
S4=HSV(:,:,2);
V4=HSV(:,:,3);

Hmin4=min(H4);
Smin4=min(S4);
Vmin4=min(V4);

Hmin4=mean(Hmin4);
Smin4=mean(Smin4);
Vmin4=mean(Vmin4);

Hmax4=max(H4);
Smax4=max(S4);
Vmax4=max(V4);

Hmax4=mean(Hmax4);
Smax4=mean(Smax4);
Vmax4=mean(Vmax4);

pelo5=imread(pelo5);
HSV = rgb2hsv(pelo5);
H5=HSV(:,:,1);
S5=HSV(:,:,2);
V5=HSV(:,:,3);

Hmin5=min(H5);
Smin5=min(S5);
Vmin5=min(V5);

Hmin5=mean(Hmin5);
Smin5=mean(Smin5);
Vmin5=mean(Vmin5);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
Hmax5=max(H5);
Smax5=max(S5);
Vmax5=max(V5);

Hmax5=mean(Hmax5);
Smax5=mean(Smax5);
Vmax5=mean(Vmax5);

pelo6=imread(pelo6);
HSV = rgb2hsv(pelo6);
H6=HSV(:,:,1);
S6=HSV(:,:,2);
V6=HSV(:,:,3);

Hmin6=min(H6);
Smin6=min(S6);
Vmin6=min(V6);

Hmin6=mean(Hmin6);
Smin6=mean(Smin6);
Vmin6=mean(Vmin6);

Hmax6=max(H6);
Smax6=max(S6);
Vmax6=max(V6);

Hmax6=mean(Hmax6);
Smax6=mean(Smax6);
Vmax6=mean(Vmax6);

pelo7=imread(pelo7);
HSV = rgb2hsv(pelo7);
H7=HSV(:,:,1);
S7=HSV(:,:,2);
V7=HSV(:,:,3);

Hmin7=min(H7);
Smin7=min(S7);
Vmin7=min(V7);

Hmin7=mean(Hmin7);
Smin7=mean(Smin7);
Vmin7=mean(Vmin7);

Hmax7=max(H7);
Smax7=max(S7);
Vmax7=max(V7);

Hmax7=mean(Hmax7);
Smax7=mean(Smax7);
Vmax7=mean(Vmax7);

pelo8=imread(pelo8);
HSV = rgb2hsv(pelo8);
H8=HSV(:,:,1);
S8=HSV(:,:,2);
V8=HSV(:,:,3);

Hmin8=min(H8);
Smin8=min(S8);
Vmin8=min(V8);

Hmin8=mean(Hmin8);
Smin8=mean(Smin8);
Vmin8=mean(Vmin8);

Hmax8=max(H8);
Smax8=max(S8);
Vmax8=max(V8);

Hmax8=mean(Hmax8);
Smax8=mean(Smax8);
Vmax8=mean(Vmax8);

a=[Hmax1, Hmax2, Hmax3, Hmax4, Hmax5, Hmax6, Hmax7, Hmax8];
Hmax=mean(a);
b=[Smax1, Smax2, Smax3, Smax4, Smax5, Smax6, Smax7, Smax8];
Smax=mean(b);
c=[Vmax1, Vmax2, Vmax3, Vmax4, Vmax5, Vmax6, Vmax7, Vmax8];
Vmax=mean(c);

d=[Hmin1, Hmin2, Hmin3, Hmin4, Hmin5, Hmin6, Hmin7, Hmin8];
Hmin=mean(d);
e=[Smin1, Smin2, Smin3, Smin4, Smin5, Smin6, Smin7, Smin8];
Smin=mean(e);
f=[Vmin1, Vmin2, Vmin3, Vmin4, Vmin5, Vmin6, Vmin7, Vmin8];
```


Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
Vmin=mean(f);

function [Hmax,Hmin, Smax,Smin, Vmax,Vmin]=
obtenerHSVPiel(piel1,piel2,piel3,piel4,piel5,piel6,piel7,piel8,piel9,piel10,piel11,piel12,piel13,piel14,piel15,piel16,piel17)

%% Carga y convierte las plantillas de piel a HSV
piel1=imread(piel1);
HSV = rgb2hsv(piel1);

H1=HSV(:,:,1);
H1m=mean(mean(H1));
S1=HSV(:,:,2);
S1m=mean(mean(S1));
V1=HSV(:,:,3);
V1m=mean(mean(V1));

piel2=imread(piel2);
HSV = rgb2hsv(piel2);

H2=HSV(:,:,1);
H2m=mean(mean(H2));
S2=HSV(:,:,2);
S2m=mean(mean(S2));
V2=HSV(:,:,3);
V2m=mean(mean(V2));

piel3=imread(piel3);
HSV = rgb2hsv(piel3);

H3=HSV(:,:,1);
H3m=mean(mean(H3));
S3=HSV(:,:,2);
S3m=mean(mean(S3));
V3=HSV(:,:,3);
V3m=mean(mean(V3));

piel4=imread(piel4);
HSV = rgb2hsv(piel4);

H4=HSV(:,:,1);
H4m=mean(mean(H4));
S4=HSV(:,:,2);
S4m=mean(mean(S4));
V4=HSV(:,:,3);
V4m=mean(mean(V4));

piel5=imread(piel5);
HSV = rgb2hsv(piel5);

H5=HSV(:,:,1);
H5m=mean(mean(H5));
S5=HSV(:,:,2);
S5m=mean(mean(S5));
V5=HSV(:,:,3);
V5m=mean(mean(V5));

piel6=imread(piel6);
HSV = rgb2hsv(piel6);

H6=HSV(:,:,1);
H6m=mean(mean(H6));
S6=HSV(:,:,2);
S6m=mean(mean(S6));
V6=HSV(:,:,3);
V6m=mean(mean(V6));

piel7=imread(piel7);
HSV = rgb2hsv(piel7);

H7=HSV(:,:,1);
H7m=mean(mean(H7));
S7=HSV(:,:,2);
S7m=mean(mean(S7));
V7=HSV(:,:,3);
V7m=mean(mean(V7));

piel8=imread(piel8);
HSV = rgb2hsv(piel8);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
H8=HSV(:,:,1);
H8m=mean(mean(H8));
S8=HSV(:,:,2);
S8m=mean(mean(S8));
V8=HSV(:,:,3);
V8m=mean(mean(V8));

piel9=imread(piel9);
HSV = rgb2hsv(piel9);

H9=HSV(:,:,1);
H9m=mean(mean(H9));
S9=HSV(:,:,2);
S9m=mean(mean(S9));
V9=HSV(:,:,3);
V9m=mean(mean(V9));

piel10=imread(piel10);
HSV = rgb2hsv(piel10);

H10=HSV(:,:,1);
H10m=mean(mean(H10));
S10=HSV(:,:,2);
S10m=mean(mean(S10));
V10=HSV(:,:,3);
V10m=mean(mean(V10));

piel11=imread(piel11);
HSV = rgb2hsv(piel11);

H11=HSV(:,:,1);
H11m=mean(mean(H11));
S11=HSV(:,:,2);
S11m=mean(mean(S11));
V11=HSV(:,:,3);
V11m=mean(mean(V11));

piel12=imread(piel12);
HSV = rgb2hsv(piel12);

H12=HSV(:,:,1);
H12m=mean(mean(H12));
S12=HSV(:,:,2);
S12m=mean(mean(S12));
V12=HSV(:,:,3);
V12m=mean(mean(V12));

piel13=imread(piel13);
HSV = rgb2hsv(piel13);

H13=HSV(:,:,1);
H13m=mean(mean(H13));
S13=HSV(:,:,2);
S13m=mean(mean(S13));
V13=HSV(:,:,3);
V13m=mean(mean(V13));

piel14=imread(piel14);
HSV = rgb2hsv(piel14);

H14=HSV(:,:,1);
H14m=mean(mean(H14));
S14=HSV(:,:,2);
S14m=mean(mean(S14));
V14=HSV(:,:,3);
V14m=mean(mean(V14));

piel15=imread(piel15);
HSV = rgb2hsv(piel15);

H15=HSV(:,:,1);
H15m=mean(mean(H15));
S15=HSV(:,:,2);
S15m=mean(mean(S15));
V15=HSV(:,:,3);
V15m=mean(mean(V15));

piel16=imread(piel16);
HSV = rgb2hsv(piel16);

H16=HSV(:,:,1);
H16m=mean(mean(H16));
S16=HSV(:,:,2);
```

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

```
S16m=mean(mean(S16));
V16=HSV(:,:,3);
V16m=mean(mean(V16));

piel17=imread(piel17);
HSV = rgb2hsv(piel17);

H17=HSV(:,:,1);
H17m=mean(mean(H17));
S17=HSV(:,:,2);
S17m=mean(mean(S17));
V17=HSV(:,:,3);
V17m=mean(mean(V17));

a=[H1m,H2m, H3m, H4m, H5m,H6m,H7m,H8m, H9m,H10m,H11m, H12m, H13m, H14m, H15m, H16m, H17m];
Hmin=min(a);
Hmax=max(a);

b=[S1m,S2m, S3m, S4m, S5m,S6m,S7m,S8m,S9m,S10m,S11m, S12m, S13m, S14m, S15m, S16m, S17m];
Smin=min(b);
Smax=max(b);

c=[V1m,V2m, V3m, V4m, V5m,V6m,V7m,V8m,V9m,V10m,V11m, V12m, V13m, V14m, V15m, V16m, V17m];
Vmin=min(c);
Vmax=max(c);
```


7. BIBLIOGRAFIA

- [1] Aguerrebere Otegui, Cecilia; Capdehourat Longres, German; Delbracio Bentancor, Mauricio y Mateu Graside, Matias. Documentación de Proyecto de Grado, Ingeniería Eléctrica Universidad de la República de Uruguay.
http://iie.fing.edu.uy/investigacion/grupos/biometria/proyectos/aguara/descargas/documenta_aguara_v1.0.pdf
[Visitado 14/09/2009]

- [2] Armengot Iborra, Marcelo: *"Análisis comparativo de métodos basados en subespacios aplicados al reconocimiento de caras"*. Universidad de Valencia, 2006.

- [3] Brand, J.; Mason, J. S. "A comparative assessment of three approaches to pixel-level human skin-detection". 15th International Conference on Pattern Recognition, 2000.

- [4] Chang, Henry; Robles, Ulises. "Face Detection" EE368 Final Project Report, 2000
<http://www-cs-students.stanford.edu/~robles/ee368/main.html>
[Visitado 02/11/2009]

- [5] "Color Space Conversión Overview -Image Processing Toolkits", Manual interactivo de Black Ice Software.

- [6] Corrales Hermoso, Alberto Luis y Sanz Montemayor, Antonio. "Sonificación de imágenes en 2-D", Proyecto Fin de Carrera de la Universidad Rey Juan Carlos, Ingeniería Técnica en Informática de Sistemas.
<http://www.escet.urjc.es/~asanz/documentos/MemoriaSonificacionJun03.pdf>
[Visitado 17/09/2009]

- [7] Cid Sueiro, Jesus; Alba, Jose Luis y Mora, Inmaculada, Universidad Carlos III de Madrid, Universidad de Vigo y Universidad Rey Juan Carlos. "Tratamiento Digital de Imagen", 2006.

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- [8] De Miguel, Darío. "Detección automática del color de la piel en imágenes bidimensionales basado en el análisis de regiones". Proyecto Fin de Carrera de Ingeniería Técnica en Informática de Sistemas, Universidad Rey Juan Carlos; 2005.
- [9] "Detección de caras y análisis de expresiones faciales".
<http://alojamientos.us.es/gtocom/pid/pid10/deteccioncaras.htm#Toc32300901>
[Visitado 19/01/2010]
- [10] Gámez Jiménez, Carmen Virginia. "Diseño y Desarrollo de un sistema de Reconocimiento de Caras". Proyecto Fin de Carrera de Ingeniería de Telecomunicaciones de Universidad Carlos III, de Madrid.
- [11] González, Rafael C. y Woods, Richard E. "Tratamiento Digital de Imágenes", ed. Addison-Wesley/Díaz de Santo, 1996
- [12] Image CLEF - Image Retrieval in CLEF.
<http://www.imageclef.org/>
[Visitado 13/5/2009]
- [13] Jure Kovac; Peter Peer; Franc Solina; "Human Skin Colour Clustering for Face Detection". Faculty of Computer and Information Science, University of Ljubljana; 2003
- [14] Prem Kuchi; Prasad Gabbur; P. Subbanna Bhat; Suman David. "Human Face Detection and Tracking using Skin Color Modeling and Connected Component Operators". IETE Journal of research, 2002.
- [15] Lee, J. Y. and Yoo, S. I. "An elliptical boundary model for skin color detection". Proceedings of the International Conference on Image Science, System and Technology; 2002
- [16] Matriz de Confusion de Wikipedia
http://es.wikipedia.org/wiki/Matriz_de_confusion
[Visitado 22/01/2010]

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- [17] Multiracial. Mezcla de razas de la Wikipedia
<http://en.wikipedia.org/wiki/Multiracial>
[Visitado 16/12/2009]
- [18] Pascal VOC. The PASCAL Visual Object Classes Homepage.
<http://pascallin.ecs.soton.ac.uk/challenges/VOC/>
[Visitado 13/05/2010]
- [19] Pérez; M. A. Vicente; C. Fernández; O. Reinoso; A. Gil; "Aplicación de los diferentes espacios de color para detección y seguimiento de caras". Departamento de Ingeniería de Sistemas Industriales de la Universidad Miguel Hernández de Elche (Alicante).
- [20] Precision and Recall. Wikipedia
http://en.wikipedia.org/wiki/Precision_and_recall
[Visitado 22/01/2010]
- [21] Proyecto de investigación Empresa DAEDALUS
<http://www.daedalus.es/i-d-i/clef-cross-language-evaluation-forum/>
[Visitado 27/01/2010]
- [22] "Robótica y Visión artificial" Apuntes de la asignatura Robótica de Ingeniería Técnica Industrial, Universidad Politécnica de Madrid.
<http://www.elai.upm.es/spain/Asignaturas/Robotica/ApuntesVA/cap4Procesadov1.pdf>
[Visitado 06/10/2009]
- [23] Roser Ballester, Miguel; Arnoso, Antonio. "Percepción computacional". Universidad Carlos III de Madrid, 2008.
- [24] Sahoo, P.k., " A survey of thresholding Techniques", Computer Vision, Graphics and Image processing, 1988.
- [25] "Tratamiento Digital de la Imagen ", Curso interactivo de la Universidad de Málaga.
<http://campusvirtual.uma.es/tdi/>
[Visitado 02/06/2009]

Sistema Automático de Detección y Etiquetado de Caras en Imágenes

- [26] "Tratamiento Digital de imágenes". Cursos básicos de Teledetección.
http://www.innovanet.com.ar/gis/TELEDETE/TELEDETE/trad.htm
[Visitado 15/09/2009]
- [27] Técnicas de Filtrado
<http://www.quesabesde.com/camdig/articulos.asp?articulo=137>
[Visitado 19/01/2010]
- [28] Terrillon, J. -C.; Shirazi, M. N.; Fukamachi, H.; Akamatsu, S. "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images". Automatic Face and Gesture Recognition, Proceedings Fourth IEEE International Conference, 2000.
- [29] Vázquez Vázquez, Nieves. "Sistema de Detección y reconocimiento de objetos en imágenes". Proyecto Fin de Carrera de Ingeniería de Telecomunicaciones, Universidad Carlos III de Madrid, 2009.
- [30] Vladimir Vezhnevets; Vassili Sazonov; Alla Andreeva. "A Survey on Pixel-Based Skin Color Detection Techniques". Machine Graphics & Vision International Journal, 2005
- [31] Wilke, Ralf. "Fundamentos de Procesamiento de Imágenes" Pontificia Universidad Católica de Chile.
http://www.ralfwilke.com/download/puc/tareal_ralf_wilke_imagenes.pdf
[Visitado 03/12/2009]
- [32] Yang, M.H; Kriegman, D.J, "Detecting Faces in Images: A Survey ,IEEE Transactions on Pattern Analysis and Machine Intelligence" .Vol. 24 y 83.
- [33] Zarit, B. D.; Super, B. J. and Quek, F. K. H. "Comparison of five color models in skin pixel classification". Recognition, Analysis and Tracking of faces and gestures in Real-Time systems, 1999.